マニュアルページセクション 1M: システム管理コマンド

Part No: E29114
2011 年 11 月

ORACLE

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

120411@25097

目次

はじめに 23

紹介 27
Intro(1M) 28

システム管理コマンド - パート 1 31
6to4relay(1M) 32
acct(1M) 35
acctadm(1M) 38
acctcms(1M) 42
acctcon(1M) 44
acctmerg(1M) 46
acctprc(1M) 47
acctsh(1M) 49
acpihpd(1M) 52
adbgen(1M) 53
add_allocatable(1M) 56
addbadsec(1M) 59
add_drv(1M) 61
arp(1M) 69
asradm(1M) 72
asr-notify(1M) 75
atohexlabel(1M) 78
audit(1M) 80
auditconfig(1M) 81
auditd(1M) 91
auditrecord(1M) 93

auditreduce(1M) 97
auditstat(1M) 106
audit_warn(1M) 108
automount(1M) 111
automountd(1M) 121
autopush(1M) 124
bart(1M) 126
beadm(1M) 132
boot(1M) 140
bootadm(1M) 161
bootconfchk(1M) 165
busstat(1M) 166
captoinfo(1M) 171
catman(1M) 172
cfgadm(1M) 176
cfgadm_ac(1M) 189
cfgadm_cardbus(1M) 194
cfgadm_fp(1M) 195
cfgadm_ib(1M) 204
cfgadm_pci(1M) 214
cfgadm_sata(1M) 220
cfgadm_sbd(1M) 229
cfgadm_scsi(1M) 244
cfgadm_sdcard(1M) 252
cfgadm_shp(1M) 257
cfgadm_sysctrl(1M) 268
cfgadm_usb(1M) 275
chat(1M) 287
check-hostname(1M) 295
check-permissions(1M) 296
chk_encodings(1M) 297
chroot(1M) 299
cimworkshop(1M) 300
clear_locks(1M) 303
clinfo(1M) 304
clri(1M) 305

consadm(1m) 306
console-reset(1M) 308
conv_lp(1M) 309
conv_lpd(1M) 310
coreadm(1M) 311
cpustat(1M) 318
croinfo(1M) 322
cron(1M) 330
cryptoadm(1M) 332
cvcd(1M) 341
datadm(1M) 343
dcs(1M) 346
dd(1M) 348
ddu(1M) 355
ddu-text(1M) 357
devchassisd(1M) 358
devfsadm(1M) 360
device_allocate(1M) 363
device_remap(1M) 364
devinfo(1M) 366
devlinks(1M) 367
devnm(1M) 371
devprop(1M) 372
df(1M) 374
dfmounts(1M) 379
dfmounts_nfs(1M) 381
dfshares(1M) 382
dfshares_nfs(1M) 383
df_ufs(1M) 385
dhcpagent(1M) 386
dhcpconfig(1M) 395
dhcpmgr(1M) 403
dhtadm(1M) 405
dig(1M) 411
directoryserver(1M) 420
disks(1M) 441

diskscan(1M) 445
dispadmin(1M) 446
dladm(1M) 449
dlmgmtd(1M) 508
dlstat(1M) 509
dmesg(1M) 517
dminfo(1M) 518
dns-sd(1M) 520
dnssec-dsfromkey(1M) 524
dnssec-keyfromlabel(1M) 526
dnssec-keygen(1M) 528
dnssec-makekeyset(1M) 531
dnssec-signkey(1M) 533
dnssec-signzone(1M) 535
domainname(1M) 540
drd(1M) 542
drvconfig(1M) 543
dsbitmap(1M) 545
dscfg(1M) 547
dscfgadm(1M) 550
dscfglockd(1M) 552
dsstat(1M) 553
dsvclockd(1M) 561
dtrace(1M) 562
dumpadm(1M) 571
editmap(1M) 576
edquota(1M) 578
eeprom(1M) 580
efdaemon(1M) 592
embedded_su(1M) 593
etrn(1M) 597
fbconfig(1M) 599
fbconf_xorg(1M) 602
fcinfo(1M) 610
fdetach(1M) 629
fdisk(1M) 630

ff(1M) 638
ff_ufs(1M) 640
fiocompress(1M) 641
flowadm(1M) 642
flowstat(1M) 651
fmadm(1M) 656
fmd(1M) 664
fmdump(1M) 667
fmstat(1M) 675
fmthard(1M) 678
format(1M) 681
fpsd(1M) 686
fruadm(1M) 687
fsck(1M) 689
fsck_pcfs(1M) 693
fsck_udfs(1M) 695
fsck_ufs(1M) 698
fsdb(1M) 701
fsdb_udfs(1M) 702
fsdb_ufs(1M) 710
fsirand(1M) 720
fssnap(1M) 721
fssnap_ufs(1M) 723
fsstat(1M) 729
fstyp(1M) 736
fuser(1M) 738
fwflash(1M) 741
fwtmp(1M) 746
getdevpolicy(1M) 747
getent(1M) 748
gettable(1M) 751
getty(1M) 752
gkadmin(1M) 754
groupadd(1M) 756
groupdel(1M) 758
groupmod(1M) 759

growfs(1M) 761
gsscred(1M) 764
gssd(1M) 766
hald(1M) 767
hal-device(1M) 768
hal-fdi-validate(1M) 769
hal-find(1M) 770
hal-get-property(1M) 771
halt(1M) 773
hextoalabel(1M) 774
host(1M) 775
hostconfig(1M) 778
hotplug(1M) 780
hotplugd(1M) 790
htable(1M) 792
ickey(1M) 793
id(1M) 794
idmap(1M) 797
idmapd(1M) 811
idsconfig(1M) 813
ifconfig(1M) 815
if_mpadm(1M) 843
ifparse(1M) 845
iiadm(1M) 847
iicpbmp(1M) 860
iicpshd(1M) 861
ikeadm(1M) 862
ikecert(1M) 871
ilbadm(1M) 886
ilbd(1M) 903
ilomconfig(1M) 905
imqadmin(1M) 907
imqbrokerd(1M) 908
imqcmd(1M) 913
imqdbmgr(1M) 926
imqkeytool(1M) 929

imqobjmgr(1M) 931
imqusermgr(1M) 940
in.chargend(1M) 943
in.comsat(1M) 944
in.daytimed(1M) 945
in.dhcpd(1M) 946
in.discardd(1M) 952
in.echod(1M) 953
inetadm(1M) 954
inetconv(1M) 958
inetd(1M) 961
in.fingerd(1M) 970
infocmp(1M) 972
in.iked(1M) 975
init(1M) 980
init.sma(1M) 986
init.wbem(1M) 987
inityp2l(1M) 989
in.lpd(1M) 991
in.mpathd(1M) 993
in.ndpd(1M) 998
in.rarpd(1M) 1001
in.rdisc(1M) 1003
in.rexecd(1M) 1005
in.ripngd(1M) 1007
in.rlogind(1M) 1010
in.routed(1M) 1014
in.rshd(1M) 1020
in.rwhod(1M) 1024
install(1M) 1026
installboot(1M) 1028
installf(1M) 1030
installgrub(1M) 1034
in.stdiscover(1M) 1036
in.stlisten(1M) 1037
in.talkd(1M) 1038

in.telnetd(1M) 1039
in.tftpd(1M) 1044
in.timed(1M) 1046
intrd(1M) 1047
intrstat(1M) 1048
in.uucpd(1M) 1051
iostat(1M) 1053
ipaddrsel(1M) 1060
ipadm(1M) 1064
ipf(1M) 1097
ipfs(1M) 1102
ipfstat(1M) 1104
ipmgmtd(1M) 1108
ipmon(1M) 1109
ipmpstat(1M) 1112
ipnat(1M) 1120
ippool(1M) 1122
ipqosconf(1M) 1125
ipsecalgs(1M) 1136
ipsecconf(1M) 1143
ipseckey(1M) 1167
iscsiadm(1M) 1182
isns(1M) 1196
isnsadm(1M) 1199
itadm(1M) 1206
itu(1M) 1215
k5srvutil(1M) 1218
kadb(1M) 1220
kadmin(1M) 1222
kadmind(1M) 1237
kcfd(1M) 1241
kclient(1M) 1242
kdb5_ldap_util(1M) 1248
kdb5_util(1M) 1259
kdcmgr(1M) 1265
kernel(1M) 1269

keyserv(1M) 1273
killall(1M) 1275
kmscfg(1M) 1276
kprop(1M) 1279
kpropd(1M) 1281
kproplog(1M) 1283
krb5kdc(1M) 1285
ksslcfg(1M) 1287
kstat(1M) 1292
ktkt_warnd(1M) 1296
labeld(1M) 1297
labelit(1M) 1298
labelit_hsfs(1M) 1300
labelit_udfs(1M) 1301
labelit_ufs(1M) 1303
latencytop(1M) 1304
ldapaddent(1M) 1307
ldap_cachemgr(1M) 1313
ldapclient(1M) 1315
ldmad(1M) 1326
link(1M) 1327
llc2_loop(1M) 1329
lldpadm(1M) 1331
lldpd(1M) 1346
lms(1M) 1347
localectr(1M) 1348
locator(1M) 1352
lockd(1M) 1353
lockfs(1M) 1356
lockstat(1M) 1360
lofiadm(1M) 1370
logadm(1M) 1378
logins(1M) 1387
lpget(1M) 1389
lpset(1M) 1391
luxadm(1M) 1393

システム管理コマンド - パート 2 1407
mail.local(1M) 1408
makedbm(1M) 1410
makemap(1M) 1412
makeuuid(1M) 1414
masfcnv(1M) 1416
mdlogd(1M) 1422
mdmonitord(1M) 1424
mdnsd(1M) 1425
medstat(1M) 1427
metaclear(1M) 1429
metadb(1M) 1432
metadevadm(1M) 1438
metahs(1M) 1441
metaimport(1M) 1445
metainit(1M) 1447
metaoffline(1M) 1458
metaparam(1M) 1460
metarecover(1M) 1462
metarename(1M) 1465
metareplace(1M) 1468
metaset(1M) 1471
metassist(1M) 1480
metastat(1M) 1485
metasync(1M) 1490
metattach(1M) 1492
mib2mof(1M) 1497
mibiisa(1M) 1499
mkbootmedia(1M) 1523
mkdevalloc(1M) 1524
mkdevmaps(1M) 1525
mkfifo(1M) 1526
mkfile(1M) 1528
mkfs(1M) 1529
mkfs_pcfs(1M) 1531
mkfs_udfs(1M) 1535

mkfs_ufs(1M) 1537
mknod(1M) 1542
mkntfs(1M) 1543
mkpwdict(1M) 1546
modinfo(1M) 1547
modload(1M) 1549
modunload(1M) 1551
mofcomp(1M) 1552
mofreg(1M) 1555
monitor(1M) 1558
mount(1M) 1570
mountall(1M) 1576
mountd(1M) 1578
mount_hsfs(1M) 1580
mount_nfs(1M) 1583
mount_pcfs(1M) 1593
mount_smbfs(1M) 1596
mount_tmpfs(1M) 1603
mount_udfs(1M) 1605
mount_ufs(1M) 1607
mpathadm(1M) 1611
mpstat(1M) 1618
msgid(1M) 1621
mvdir(1M) 1622
named(1M) 1623
named-checkconf(1M) 1629
named-checkzone(1M) 1630
ncaconfd(1M) 1633
ncheck(1M) 1634
ncheck_ufs(1M) 1636
ndd(1M) 1637
ndmpadm(1M) 1639
ndmpd(1M) 1643
ndmpstat(1M) 1644
netadm(1M) 1647
netcfg(1M) 1652

netcfgd(1M) 1668
netservices(1M) 1669
netstat(1M) 1670
netstrategy(1M) 1682
newaliases(1M) 1683
newfs(1M) 1685
newkey(1M) 1690
nfs4cbd(1M) 1691
nfsd(1M) 1692
nfslogd(1M) 1696
nfsmapid(1M) 1699
nfsref(1M) 1702
nfsstat(1M) 1704
nscadm(1M) 1711
nscd(1M) 1713
nscfg(1M) 1716
nslookup(1M) 1720
nsupdate(1M) 1724
ntfscat(1M) 1729
ntfsclone(1M) 1731
ntfscluster(1M) 1736
ntfscmp(1M) 1738
ntfscp(1M) 1739
ntfsfix(1M) 1741
ntfsinfo(1M) 1742
ntfslabel(1M) 1744
ntfsls(1M) 1746
ntfsprogs(1M) 1748
ntfsresize(1M) 1750
ntfsundelete(1M) 1754
nwamd(1M) 1759
obpsym(1M) 1760
oplhpd(1M) 1762
parted(1M) 1763
pbind(1M) 1767
pcitool(1M) 1770

pginfo(1M) 1774
pgstat(1M) 1779
picld(1M) 1785
ping(1M) 1787
pkg2du(1M) 1792
pkgadd(1M) 1794
pkgadm(1M) 1801
pkgask(1M) 1806
pkgchk(1M) 1808
pkgcond(1M) 1812
pkgrm(1M) 1814
plockstat(1M) 1817
pntadm(1M) 1819
polkit-is-privileged(1M) 1826
pooladm(1M) 1827
poolbind(1M) 1830
poolcfg(1M) 1832
poold(1M) 1837
poolstat(1M) 1839
ports(1M) 1843
poweradm(1M) 1847
powertop(1M) 1851
pppd(1M) 1854
pppoec(1M) 1882
pppoed(1M) 1885
pppstats(1M) 1890
praudit(1M) 1893
projadd(1M) 1895
projdel(1M) 1898
projmod(1M) 1900
prstat(1M) 1905
prtconf(1M) 1913
prtdiag(1M) 1916
prtdscp(1M) 1918
prtfru(1M) 1920
prtpicl(1M) 1921

prtvtoc(1M) 1922
psradm(1M) 1925
psrinfo(1M) 1928
psrset(1M) 1931
pwck(1M) 1936
pwconv(1M) 1937
quot(1M) 1939
quota(1M) 1941
quotacheck(1M) 1942
quotaon(1M) 1944
raidctl(1M) 1946
ramdiskadm(1M) 1957
rcapadm(1M) 1959
rcapd(1M) 1962
rctladm(1M) 1964
rdate(1M) 1966
reboot(1M) 1967
rem_drv(1M) 1971
remove_allocatable(1M) 1973
removef(1M) 1975
reparsed(1M) 1977
repquota(1M) 1978
rmmount(1M) 1979
rmt(1M) 1981
rmvolmgr(1M) 1983
rndc(1M) 1985
rndc-confgen(1M) 1987
roleadd(1M) 1989
roledel(1M) 1994
rolemod(1M) 1996
root_archive(1M) 2001
route(1M) 2002
routeadm(1M) 2011
rpcbind(1M) 2016
rpc.bootparamd(1M) 2019
rpcinfo(1M) 2020

rpc.mdcommd(1M) 2024
rpc.metad(1M) 2025
rpc.metamedd(1M) 2026
rpc.metamhd(1M) 2027
rpc.rexd(1M) 2028
rpc.rstatd(1M) 2030
rpc.rusersd(1M) 2031
rpc.rwalld(1M) 2032
rpc.smserverd(1M) 2033
rpc.sprayd(1M) 2034
rpc.yppasswdd(1M) 2035
rpc.ypupdated(1M) 2038
rquotad(1M) 2039
rsh(1M) 2040
rtc(1M) 2042
rtquery(1M) 2043
runacct(1M) 2045
rwall(1M) 2048
sar(1M) 2049
sasinfo(1M) 2051
savecore(1M) 2062
sbdadm(1M) 2064
scadm(1M) 2067
sckmd(1M) 2076
scmadm(1M) 2077
sconadm(1M) 2079
sdpadm(1M) 2084
sendmail(1M) 2085
sf880drd(1M) 2112
sftp-server(1M) 2113
shadowd(1M) 2115
shadowstat(1M) 2116
share(1M) 2117
shareall(1M) 2119
sharectl(1M) 2120
share_nfs(1M) 2123

share_smb(1M) 2130
showmount(1M) 2136
shutdown(1M) 2137
slpd(1M) 2140
smattrpop(1M) 2142
smbadm(1M) 2146
smbd(1M) 2154
smbiod(1M) 2156
smbios(1M) 2157
smbstat(1M) 2160
smrsh(1M) 2164
smtnrhdb(1M) 2165
smtnrhtp(1M) 2170
smtnzonecfg(1M) 2175
smtp-notify(1M) 2180
sndradm(1M) 2183
sndrd(1M) 2193
sndrsyncd(1M) 2195
snmpdx(1M) 2197
snmp-notify(1M) 2200
snmpXwbemd(1M) 2202
snoop(1M) 2205
soconfig(1M) 2218
soladdapp(1M) 2221
soldelapp(1M) 2222
solstice(1M) 2223
sppptun(1M) 2224
spray(1M) 2227
srptadm(1M) 2228
sshd(1M) 2232
ssh-keysign(1M) 2248
statd(1M) 2250
stclient(1M) 2252
stmfadm(1M) 2257
stmsboot(1M) 2268
strace(1M) 2273

strclean(1M) 2275
strerr(1M) 2276
sttydefs(1M) 2278
su(1M) 2280
sulogin(1M) 2283
svadm(1M) 2284
svcadm(1M) 2286
svccfg(1M) 2294
svc.configd(1M) 2307
svc.ipfd(1M) 2308
svc.startd(1M) 2314
swap(1M) 2321
sync(1M) 2325
syncinit(1M) 2326
syncloop(1M) 2329
syncstat(1M) 2332
sysconfig(1M) 2335
sysdef(1M) 2339
syseventadm(1M) 2341
syseventconfd(1M) 2346
syseventd(1M) 2347
syslogd(1M) 2349
tapes(1M) 2353
tcxconfig(1M) 2357
th_define(1M) 2358
th_manage(1M) 2369
tic(1M) 2371
tncfg(1M) 2372
tnchkdb(1M) 2379
tnctl(1M) 2381
tnd(1M) 2384
tninfo(1M) 2386
tpmadm(1M) 2388
traceroute(1M) 2391
trapstat(1M) 2398
ttymon(1M) 2409

tunefs(1M) 2414
txzonemgr(1M) 2416
tzreload(1M) 2418
tzselect(1M) 2419
uadmin(1M) 2420
ucodeadm(1M) 2421
ufsdump(1M) 2423
ufsrestore(1M) 2430
unshare(1M) 2437
unshare_nfs(1M) 2438

システム管理コマンド - パート **3** 2439
update_drv(1M) 2440
useradd(1M) 2444
userdel(1M) 2451
usermod(1M) 2453
utmpd(1M) 2460
uucheck(1M) 2462
uucico(1M) 2463
uucleanup(1M) 2465
uusched(1M) 2467
Uutry(1M) 2468
uuxqt(1M) 2469
vbiosd(1M) 2470
vdiskadm(1M) 2471
virt-convert(1M) 2481
virtinfo(1M) 2484
vmstat(1M) 2487
vntsd(1M) 2491
volcopy(1M) 2495
volcopy_ufs(1M) 2497
vrrpadm(1M) 2498
vrrpd(1M) 2503
vscanadm(1M) 2504
vscand(1M) 2510

vtdaemon(1M) 2512
wall(1M) 2513
wanboot_keygen(1M) 2515
wanboot_keymgmt(1M) 2517
wanboot_p12split(1M) 2518
wanbootutil(1M) 2519
wbemadmin(1M) 2520
wbemconfig(1M) 2523
wbemlogviewer(1M) 2524
wcadmin(1M) 2526
whodo(1M) 2531
wpad(1M) 2533
wracct(1M) 2535
wusbadm(1M) 2537
ypbind(1M) 2544
ypinit(1M) 2547
ypmake(1M) 2549
ypmap2src(1M) 2551
yppoll(1M) 2553
yppush(1M) 2554
ypserv(1M) 2556
ypserv_resolv(1M) 2560
ypset(1M) 2561
ypstart(1M) 2563
ypxfr(1M) 2564
zdb(1M) 2566
zdump(1M) 2567
zfs(1M) 2568
zfs_allow(1M) 2610
zfs_encrypt(1M) 2616
zfs_share(1M) 2625
zic(1M) 2633
zoneadm(1M) 2638
zoneadmd(1M) 2647
zonecfg(1M) 2648
zonep2vchk(1M) 2675

zonestatd(1M) 2688
zpool(1M) 2690
zstreamdump(1M) 2718

はじめに

SunOS オペレーティングシステムの初心者でも、熟練したユーザーでも、オンラインのマニュアルページを使ってシステムおよびその機能に関する情報を入手できます。マニュアルページは、「それは何を行うのか」という質問に簡潔に回答するために作成されています。マニュアルページは一般にリファレンスマニュアルとなります。チュートリアルとしては作成されていません。

概要

次に、マニュアルページの各セクションと、それらのセクションで示される情報の概要を示します。

- セクション 1 では、オペレーティングシステムで使用できるコマンドをアルファベット順に説明しています。
- セクション 1M では、主にシステムの保守と管理のために使用されるコマンドをアルファベット順に説明しています。
- セクション 2 では、すべてのシステムコールについて説明しています。これらの呼び出しのほとんどは、1 つ以上のエラーを返します。エラー状態は、ほかの場合には返されない戻り値によって示されます。
- セクション 3 では、セクション 2 で説明されている UNIX システムプリミティブを直接呼び出す関数を除き、さまざまなライブラリにある関数について説明しています。
- セクション 4 では、各種ファイルの形式について説明しています。該当する場合は、ファイル形式を C 言語の構造体の宣言を使用して示します。
- セクション 5 では、文字セットテーブルなど、その他各種の情報について説明しています。
- セクション 7 では、特定のハードウェア周辺装置やデバイスドライバを参照する、さまざまな特殊ファイルについて説明しています。STREAMS ソフトウェアドライバ、モジュール、およびシステムコールの STREAMS 汎用セットについても説明します。
- セクション 9E では、開発者がデバイスドライバに組み込むことができる、DDI (デバイスドライバインタフェース)/DKI (ドライバ/カーネルインタフェース) 両用、DDI 専用、および DKI 専用のエントリポイントルーチンについて説明しています。

- セクション 9F では、デバイスドライバで使用できるカーネル関数について説明しています。
- セクション 9S では、ドライバとカーネルの間で情報を共有するためにドライバで使用されるデータ構造について説明しています。

マニュアルページの一般形式は次のとおりです。各マニュアルセクションのマニュアルページは、通常この順序に従っていますが、必要な項目だけを含んでいます。たとえば、記述すべきバグがコマンドにない場合、「使用上の留意点」というセクションはありません。各セクションの詳細については `intro` ページ、マニュアルページの一般的な情報については `man(1)` を参照してください。

名前
: このセクションでは、コマンドまたは関数の名前および機能の概略を示します。

形式
: このセクションでは、コマンドまたは関数の構文を示します。コマンドやファイルが標準パスに存在しない場合は、そのフルパス名が示されます。別の引数順序が必要な場合を除いて、オプションと引数はアルファベット順に、まず 1 文字の引数、次に引数付きのオプションが記述されます。

  このセクションでは、次の特殊文字が使用されます。

  [ ]
  : 角括弧。角括弧に囲まれたオプションや引数は省略できます。角括弧がない場合、その引数は必須です。

  . . .
  : 省略記号。前の引数に複数の値を指定すること、または前の引数を複数回指定することができます (例: “`filename...`”)。

  |
  : 区切り文字。この文字で区切られている引数のうち 1 つだけを一度に指定できます。

  { }
  : 中括弧。中括弧に囲まれたオプションや引数は相互に依存しているため、括弧内のすべてを一体として扱う必要があります。

プロトコル
: このセクションは、プロトコルが記述されているファイルを示すためにサブセクション 3R でのみ使用されます。

機能説明
: このセクションでは、サービスの機能と動作を定義します。つまり、コマンドの機能を簡潔に説明します。オプションの説明や使用例はここでは示されていません。対話形式のコマンド、サブコマンド、要求、マクロ、および関数については「使用法」で説明します。

IOCTL
: このセクションは、セクション 7 のページだけに含まれます。`ioctl(2)` システムコールに適切なパラメータを提供するデバイスクラスのみが `ioctl` と呼ばれ、独自の見出しを生成しま

す。特定のデバイスに関する ioctl 呼び出しは、(そのデバイスのマニュアルページに) アルファベット順に記述されています。ioctl 呼び出しは、mtio(7I) のように、すべて io で終わる特定クラスのデバイスに使用されます。

オプション
このセクションでは、コマンドオプションの一覧と各オプションの機能のサマリーを示します。オプションは、「形式」のセクションで示されている順序で、文字通りに記述されています。オプションに引数がある場合はそのオプションの下で説明され、該当する場合はデフォルト値が示されます。

オペランド
このセクションでは、コマンドのオペランドの一覧を示し、各オペランドがコマンドの動作にどのように影響を及ぼすかを説明します。

出力
このセクションでは、コマンドによって生成される出力 (標準出力、標準エラー、または出力ファイル) について説明します。

戻り値
値を返す関数のマニュアルページの場合、このセクションには、これらの値の一覧および値が返される条件が示されます。関数が 0 や -1 などの定数値だけを返す場合は、これらの値がタグ付き段落で一覧表示されます。それ以外の場合は、各関数の戻り値が 1 つの段落で説明されます。void と宣言された関数は値を返さないため、「戻り値」では扱われません。

エラー
エラー発生時、ほとんどの関数はエラーコードを大域変数 errno に格納し、エラーの理由を示します。このセクションでは、関数が生成しうるすべてのエラーコードの一覧をアルファベット順で示し、各エラーの発生条件について説明します。複数の条件が同じエラーの原因になる場合、そのエラーコードの下に各条件が別々の段落で記述されます。

使用法
このセクションでは、詳細な説明を必要とする特殊な規則、機能、およびコマンドを示します。次に示すサブセクションは、組み込まれている機能を説明するために使用されます。

コマンド
修飾子
変数
式
入力文法

使用例
このセクションでは、コマンドまたは関数の使用例または使用方法を説明します。可能な場合は、コマンド行の入力とマシンの応答を含む完全な例が示されます。例の中で、プロンプトは example% と表示されるか、スーパーユーザーでなければならな

い場合は example# と表示されます。例のあとに、説明、変数置換規則、または戻り値が示されます。ほとんどの例が、「形式」、「機能説明」、「オプション」、「使用法」のセクションの概念を示す実例となっています。

環境変数　このセクションでは、コマンドまたは関数が影響を与えるすべての環境変数の一覧を示し、その影響について簡単に説明します。

終了ステータス　このセクションでは、コマンドが呼び出し元のプログラムまたはシェルに返す値の一覧と、これらの値が返される条件を示します。通常、正常終了には 0 が返され、0 以外の値はそれぞれのエラー状態を示します。

ファイル　このセクションでは、マニュアルページが参照するファイル、関連ファイル、およびコマンドが作成または必要とするファイルの一覧を示します。それぞれのあとに、サマリーまたは説明があります。

属性　このセクションでは、属性タイプとその対応する値を定義することにより、コマンド、ユーティリティー、およびデバイスドライバの特性を示します。詳細については、attributes(5) を参照してください。

関連項目　このセクションでは、関連するマニュアルページ、当社のマニュアル、および外部の出版物を示します。

診断　このセクションでは、診断メッセージの一覧を示し、エラーの発生条件について簡単に説明します。

警告　このセクションでは、作業環境に重大な影響を与える可能性のある特殊な条件に関する警告の一覧を示します。これは診断の一覧ではありません。

注意事項　このセクションでは、このマニュアルページの他のどのセクションにも記載されない追加情報を示します。ユーザーへの参考情報として、特に関心を引く事項を扱います。ここで重要な情報が扱われることはありません。

使用上の留意点　このセクションでは、既知の問題について説明し、可能な場合は回避方法を示します。

参 照

紹介

名前

Intro, intro – 管理コマンドおよびアプリケーションプログラムの序章

機能説明

本セクションでは、主にシステムの保守や管理に使用するコマンドを、アルファベット順に説明します。

コマンドが仮想ファイルシステムのアーキテクチャに合わせて再構成されているため、同じ名前で始まる複数のマニュアルページが存在します。たとえば、`mount` の名前に関して
は、`mount(1M)`、`mount_hsfs(1M)`、`mount_nfs(1M)`、`mount_tmpfs(1M)`、および `mount_ufs(1M)` のように 5 つが存在します。このような場合、最初にだけ、その総称コマンドの構文およびオプションが説明されています。つまり、これらのオプションは、すべてのファイルシステムのタイプに適用されるということです。以降には、そのコマンドの機能のうちファイルシステムタイプに特有な部分が説明されています。このようなマニュアルページ名には、下線 (_) とそのコマンドに関係するファイルシステムのタイプ名が伴われています。管理者は、このようなファイルシステムに特有な部分を直接呼び出してはいけません。総称コマンドは、すべてのファイルシステムに共通なインタフェースを提供します。ですから、ファイルシステムタイプに特有なマニュアルページは、個々のコマンドを説明していると考えるべきではなく、コマンドのファイルシステムに特有な面を詳細に述べたものだと考えるべきです。

コマンドの構文

特に説明しないかぎり、本セクションで説明するコマンドは、以下の構文に従って、オプションやその他の引数を受け入れます。

name [*option*(*s*)] [*cmdarg*(*s*)]

出力形式の説明

name	実行可能ファイルの名前です。
option	– *noargletter*(*s*) または – *argletter*< >*optarg* < > は、スペース (オプション)。
noargletter	引数が必要でないオプション 1 文字を表します。
argletter	引数が必要なオプション 1 文字を表します。
optarg	*argletter* に必要な引数 (文字列) です。
cmdarg	パス名 (または他のコマンドの引数)。– だけを指定すると標準入力を表します。

属性

このセクション中にリストされている属性については、`attributes(5)`を参照してください。

謝辞

Oracle America, Inc. gratefully acknowledges The Open Group for permission to reproduce portions of its copyrighted documentation. Original documentation from The Open Group can be obtained online at http://www.opengroup.org/bookstore/.

The Institute of Electrical and Electronics Engineers and The Open Group, have given us permission to reprint portions of their documentation.

In the following statement, the phrase “this text" refers to portions of the system documentation.

Portions of this text are reprinted and reproduced in electronic form in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2004 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between these versions and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html.

This notice shall appear on any product containing this material.

関連項目

getopt(1), getopt(3C), attributes(5)

診断

終了時、すべてのコマンドは、正常に終了すると 0 を返します。ゼロでない値を返した場合、間違った引数を指定した、不良で受け入れることができないデータを指定した、その他、現在はうまく処理できないなどの障害を示します。このような値は、「終了コード」、「終了ステータス」、「リターンコード」などさまざまな呼ばれ方をします。そして、特別な使い方がある場合にかぎって説明されます。

注意事項

すべてのコマンドにおいて標準の構文に準拠しているわけではありません。

参 照

システム管理コマンド - パート 1

名前

6to4relay – administer configuration for 6to4 relay router communication

形式

```
/usr/sbin/6to4relay

/usr/sbin/6to4relay [-e] [-a addr]

/usr/sbin/6to4relay [-d]

/usr/sbin/6to4relay [-h]
```

機能説明

The `6to4relay` command is used to configure 6to4 relay router communication. Relay router communication support is enabled by setting the value of a variable that stores an IPv4 address within the `tun` module. This variable is global to all tunnels and defines the policy for communication with relay routers. By default, the address is set to `INADDR_ANY` (`0.0.0.0`), and the kernel interprets the value to indicate that support for relay router communication is disabled. Otherwise, support is enabled, and the specified address is used as the IPv4 destination address when packets destined for native IPv6 (non-6to4) hosts are sent through the 6to4 tunnel interface. The `6to4relay` command uses a project private ioctl to set the variable.

`6to4relay` used without any options outputs the current, in-kernel, configuration status. Use the `-a` option to send packets to a specific relay router's `unicast` address instead of the default `anycast` address. The address specified with the `-a` option does not specify the policy for receiving traffic from relay routers. The source relay router on a received packet is non-deterministic, since a different relay router may be chosen for each sending native IPv6 end-point.

Configuration changes made by using the `6to4relay` are not persistent across reboot. The changes will persist in the kernel only until you take the tunnel down

オプション

The `6to4relay` command supports the following options:

`-a` *addr* Use the specified address, *addr*.

`-e` Enable support for relay router. Use `-a` *addr* if it is specified. Otherwise, use the default `anycast` address, 192.88.99.1.

`-d` Disable support for the relay router.

`-h` Help

オペランド

The following operands are supported:

addr A specific relay router's unicast address. *addr* must be specified as a dotted decimal representation of an IPv4 address. Otherwise, an error will occur, and the command will fail.

使用例

例 1 Printing the In-Kernel Configuration Status

Use `/usr/sbin/6to4relay` without any options to print the in-kernel configuration status.

```
example# /usr/sbin/6to4relay
```

例 1 Printing the In-Kernel Configuration Status (続き)

If 6to4 relay router communication is disabled, the administrator will see the following message:

```
6to4relay: 6to4 Relay Router communication support is disabled.
```

If 6to4 router communication is enabled, the user will see this message:

```
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 destination address of Relay Router = 192.88.99.1
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/usr/sbin/6to4relay` The default installation root

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`ifconfig(1M)`, `attributes(5)`

Huitema, C. RFC 3068, An Anycast Prefix for 6to4 Relay Routers. Network Working Group. June, 2001.

Carpenter, B. and Moore, K. RFC 3056, Connection of IPv6 Domains via IPv4 Clouds. Network Working Group. February, 2001.

診断

The `6to4relay` reports the following messages:

```
6to4relay: input (0.0.0.0) is not a valid IPv4 unicast address
```

例: The following example provides an incorrect unicast address.

```
example# 6to4relay -e -a 0.0.0.0
```

説明: The address specified with the `-a` option must be a valid unicast address.

```
6to4relay: option requires an argument —a
usage:
    6to4relay
    6to4relay -e [-a <addr>]
```

```
 6to4relay -d
 6to4relay -h
```

例: The following example does not include an argument for the -a option.

```
example# 6to4relay -e -a
```

説明: The -a option requires an argument.

```
usage:
    6to4relay
    6to4relay -e [-a <addr>]
    6to4relay -d
    6to4relay -h
```

例: The following example specifies options that are not permitted.

```
example# 6to4relay -e -d
```

説明: The options specified are not permitted. A usage message is output to the screen.

```
usage:
    6to4relay
    6to4relay -e [-a <addr>]
    6to4relay -d
    6to4relay -h
```

例: The following example specifies the -a option without specifying the -e option.

```
example# 6to4relay -a 1.2.3.4
```

説明: The -e option is required in conjunction with the -a option. A usage message is output to the screen.

```
6to4relay: ioctl (I_STR) : Invalid argument
```

例: The following example specifies an invalid address.

```
example# 6to4relay -e -a 239.255.255.255
```

説明: The address specified with the -a option must not be a class d *addr*.

名前

acct, acctdisk, acctdusg, accton, acctwtmp, closewtmp, utmp2wtmp – アカウンティングおよびいろいろなアカウントコマンドの概要

形式

```
/usr/lib/acct/acctdisk

/usr/lib/acct/acctdusg [-u filename] [-p filename]

/usr/lib/acct/accton [filename]

/usr/lib/acct/acctwtmp reason filename

/usr/lib/acct/closewtmp

/usr/lib/acct/utmp2wtmp
```

機能説明

アカウンティングソフトウェアは、アカウンティングシステムを構築するためのツール群です (C 言語のプログラムおよびシェルプロシージャから構成される)。acctsh(1M) では、C 言語のプログラムのいちばん上に組むシェルプロシージャー群を説明します。

接続時間のアカウンティングは記録を /var/adm/wtmpx (utmpx(4) を参照) に書き込むプログラム群により管理されます。acctcon(1M) では、このファイルをセッションおよび課金記録に変換するプログラムを説明します。また、acctmerg(1M) では、この課金記録について集約します。

プロセスアカウンティングは、システムカーネルによって行われます。プロセスを終了すると、1 つのプロセスにつき 1 つのレコードが、あるファイル (通常は、/var/adm/pacct) に書き込まれます。acctprc(1M) では、課金のためにこのデータを集約します。コマンドの使用状況を集約するには、acctcms(1M) を使用します。acctcom(1) を用いて、現在のプロセスデータを調査することができます。

acctmerg (acct.h(3HEAD) の tacct フォーマットを参照) を使うと、プロセスのアカウンティングレコードおよび接続時間のアカウンティングレコード (あるいは acct.h(3HEAD) で説明する tacct フォーマットのアカウンティングレコードのいずれか) をアカウンティングレコードの合計にマージしたり、集約したりすることができます。どのアカウンティングレコードにも、あるいはすべてのアカウンティングレコードにでも prtacct (acctsh(1M) を参照) をフォーマットに使用します。

acctdisk は、ユーザー ID、ログイン名およびディスクブロック数がある行を読み取り、ほかのアカウンティングレコードとマージすることができるアカウンティングレコードの合計に、それらの情報を変換します。入力ファイルが壊れているか、または正しくフォーマットされていない場合は、acctdisk はエラーを返します。

acctdusg は、その標準入力を (通常は、find / -print から) 読み取り、ログインによるディスクのリソース消費 (間接ブロックを含む) を計算します。

accton では、引数がない場合、プロセスアカウンティングをオフにします。*filename* を指定する場合は、そのファイルは、カーネルがプロセスのアカウンティングレコード (acct(2) および acct.h(3HEAD) を参照) を追加するための、既存のファイル名である必要があります。

acctwtmp は、utmpx(4) のレコードを *filename* へ書き込みます。レコードには、現在の時間および *reason* を説明する文字列が入っています。ACCOUNTING のレコードの型が割り当てられます (utmpx(4) を参照)。*reason* には、11 文字以下の文字列、数、$、またはスペースを指定してください。たとえば、以下は、リブートのプロシージャおよびシャットダウンのプロシージャでの使用例です。

```
acctwtmp "acctg on" /var/adm/wtmpx
acctwtmp "acctg off" /var/adm/wtmpx
```

現在ログオンしている各ユーザーについて、closewtmp は、偽の DEAD_PROCESS レコードを /var/adm/wtmpx ファイルに入れます。runacct (runacct(1M) を参照) がこの偽の DEAD_PROCESS を使用することにより、接続アカウンティングプロシージャーは runacct を起動する前にログオンしたユーザーが使用した時間を追跡することができます。

現在ログオンしている各ユーザーについて、runacct は utmp2wtmp を使用して、runacct が作成したファイル /var/adm/wtmpx にエントリを作成します。この /var/adm/wtmpx のエントリを見れば、引き続いて起こる runacct の呼び出しで、現在ログインしているユーザーの接続時間がわかります。

オプション

次のオプションがサポートされています。

-u *filename*　だれにも課金されないファイル名から成るレコードを、*filename* に入れます (ディスクの課金を拒否しようとしたユーザーを見つけるための潜在的な情報源となる)。

-p *filename*　*p* は、パスワードファイル filename を指定します。パスワードファイルが /etc/passwd である場合は、このオプションは不要です。

環境変数

LC_* 変数 (LC_TYPE、LC_MESSAGES、LC_TIME、LC_COLLATE、LC_NUMERIC、LC_MONETARY) (environ(5) 参照) のいずれも環境に設定されていなければ、それぞれ対応するロケールのカテゴリにおける acct の動作は、環境変数 LANG によって決定されます。もし、LC_ALL が設定されていれば、その内容が LANG 変数やその他の LC_* 変数より優先されます。前述の変数が環境にまったく設定されていなければ、C ロケール (米国スタイル) が acct の動作を決定します。

LC_CTYPE　acct の文字の処理方法を決定します。LC_CTYPE に有効な値が設定されていると、acct は、そのロケールにあった文字を含むテキストやファイル名を表示および処理できます。acct は拡張 UNIX コード (EUC) も表示および処理できます。この場合、文字は 1 バイト幅、2 バイト幅、3 バイト幅のいずれも使用できます。また、acct は 1、2、またはそれ以上のカラム幅の EUC 文字も処理することができます。C ロケールにおいては、ISO 8859-1 の文字だけが有効です。

LC_TIME　acct の日付および時間のフォーマットの処理方法を決定します。C ロケールにおいては、日付および時間の処理方法は米国ルールに従います。

ファイル

/etc/passwd	ログイン名からユーザー ID への変換に用いる
/usr/lib/acct	本マニュアルの 1M 章に含まれるアカウンティングコマンドが置かれる
/var/adm/pacct	現在のプロセスアカウンティングファイル
/var/adm/wtmpx	ユーザーアクセスまたは管理情報の履歴

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/accounting/legacy-accounting

関連項目

acctcom(1), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct.h(3HEAD), passwd(4), utmpx(4), attributes(5), environ(5)

『Oracle Solaris の管理: 一般的なタスク』

名前

acctadm – configure extended accounting facility

形式

```
/usr/sbin/acctadm [-DEsrux] [-d resource_list]
     [-e resource_list] [-f filename]
     [task | process | flow | net]
```

機能説明

`acctadm` configures various attributes of the extended accounting facility. Without arguments, `acctadm` displays the current status of the extended accounting facility.

オプション

The following options are supported:

`-d` *resource_list*
: Disable reporting of resource usage for resource. Specify *resource_list* as a comma-separated list of resources or resource groups.

  This option requires an operand. See `OPERANDS`.

`-D`
: Disable accounting of the given operand type without closing the accounting file. This option can be used to temporarily stop writing accounting records to the accounting file without closing it. To close the file use the `-x` option. See `-x`.

`-e` *resource_list*
: Enable reporting of resource usage for resource. Specify *resource_list* as a comma-separated list of resources or resource groups.

  This option requires an operand. See `OPERANDS`.

`-E`
: Enable accounting of the given operand type without sending the accounting output to a file. This option requires an operand. See `OPERANDS`.

`-f` *filename*
: Send the accounting output for the given operand type to *filename*. If *filename* exists, its contents must be of the given accounting type.

  This option requires an operand. See `OPERANDS`.

`-r`
: Display available resource groups.

  When this option is used with an operand, it displays resource groups available for a given accounting type. When no operand is specified, this option displays resource groups for all available accounting types. See OPERANDS.

`-s`
: Start method for the `smf(5)` instance. This option is used to restore the extended accounting configuration at boot.

`-x`
Deactivate accounting of the given operand type. This option also closes the accounting file for the given accounting type if it is currently open.

This option requires an operand. See `OPERANDS`.

オペランド

The `-d`, `-D`, `-e`, `-E`, `-f`, and `-x` options require an operand.

The following operands are supported:

`process`
Run `acctadm` on the process accounting components of the extended accounting facility.

`task`
Run `acctadm` on the task accounting components of the extended accounting facility.

`flow`
Run `acctadm` on the IPQoS accounting components of the extended accounting facility.

`net`
Run `acctadm` on links and flows administered by `dladm(1M)` and `flowadm(1M)`, respectively. Basic network accounting relates only to links, while extended network accounting includes both link and flow accounting.

The optional final parameter to `acctadm` represents whether the command should act on the process, system task, IPQoS, or network accounting components of the extended accounting facility.

使用例

例 1 Displaying the Current Status

The following command displays the current status. In this example, system task accounting is active and tracking only CPU resources. Process and flow accounting are not active.

```
$ acctadm
            Task accounting: active
       Task accounting file: /var/adm/exacct/task
     Tracked task resources: extended
   Untracked task resources: host
         Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
Untracked process resources: extended,host
            Flow accounting: inactive
       Flow accounting file: none
     Tracked flow resources: none
   Untracked flow resources: extended
             Net accounting: inactive
        Net accounting file: none
      Tracked Net resources: none
    Untracked Net resources: extended
```

例 2 Activating Basic Process Accounting

The following command activates basic process accounting:

```
$ acctadm -e basic -f /var/adm/exacct/proc process
```

例 3 Displaying Available Resource Groups

The following command displays available resource groups:

```
$ acctadm -r
  process:
  extended pid,uid,gid,cpu,time,command,tty,projid, \
  taskid,ancpid,wait-status,zone,flag,memory,mstate
  basic    pid,uid,gid,cpu,time,command,tty,flag
  task:
  extended taskid,projid,cpu,time,host,mstate,anctaskid,zone
  basic    taskid,projid,cpu,time
  flow:
  extended saddr,daddr,sport,dport,proto,dsfield,nbytes,npkts, \
  action,ctime,lseen,projid,uid
  basic    saddr,daddr,sport,dport,proto,nbytes,npkts,action
  net:
  extended name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
  priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpkts \
  oerrpkts,saddr,daddr,sport,dport,protocol,dsfield
  basic    name,devname,edest,vlan_tpid,vlan_tci,sap,cpuid, \
  priority,bwlimit,curtime,ibytes,obytes,ipkts,opks,ierrpkts \
  oerrpkts
```

In the output above, the lines beginning with `extended` are shown with a backslash character. In actual `acctadm` output, these lines are displayed as unbroken, long lines.

例 4 Displaying Resource Groups for Task Accounting

The following command displays resource groups for task accounting:

```
$ acctadm -r task
  extended taskid,projid,cpu,time,host,mstate,anctaskid,zone
  basic    taskid,projid,cpu,time
```

終了ステータス

The following exit values are returned:

`0`

Successful completion.

The modifications to the current configuration were valid and made successfully.

`1`

An error occurred.

A fatal error occured either in obtaining or modifying the accounting configuration.

2
Invalid command line options were specified.

95
A fatal, non-configuration error occurred during the start of the smf(5) service instance.

96
A fatal configuration error occurred during the start of the smf(5) service instance.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

dladm(1M), flowadm(1M), acct(2), attributes(5), smf(5), ipqos(7ipp)

注意事項

Both extended accounting and regular accounting can be active.

Available resources can vary from system to system, and from platform to platform.

Extended accounting configuration is stored in the service management facility (smf(5)) repository. The configuration is restored at boot by a transient service instance, one per accounting type:

```
svc:/system/extended-accounting:flow        Flow accounting
svc:/system/extended-accounting:process     Process accounting
svc:/system/extended-accounting:task        Task accounting
svc:/system/extended-accounting:net         Network accounting
```

The instances are enabled or disabled by acctadm as needed. Configuration changes are made using acctadm; service properties should not be modified directly using svccfg(1M).

Users can manage extended accounting (start accounting, stop accounting, change accounting configuration parameters) if they have the appropriate RBAC Rights profile for the accounting type to be managed:

- Extended Accounting Flow Management
- Extended Accounting Process Management
- Extended Accounting Task Management
- Extended Accounting Network Management

The preceding profiles are for, respectively, flow accounting, process accounting, task accounting, and network accounting.

名前

acctcms – command summary from process accounting records

形式

```
/usr/lib/acct/acctcms [-a [-o] [-p]] [-c] [-j] [-n] [-s]
     [-t] filename...
```

機能説明

acctcms reads one or more *filenames*, normally in the form described in acct.h(3HEAD). It adds all records for processes that executed identically named commands, sorts them, and writes them to the standard output, normally using an internal summary format.

オプション

-a Print output in ASCII rather than in the internal summary format. The output includes command name, number of times executed, total kcore-minutes, total CPU minutes, total real minutes, mean size (in K), mean CPU minutes per invocation, "hog factor," characters transferred, and blocks read and written, as in acctcom(1). Output is normally sorted by total kcore-minutes.

Use the following options only with the -a option:

-o Output a (non-prime) offshift-time-only command summary.

-p Output a prime-time-only command summary.

When -o and -p are used together, a combination prime-time and non-prime-time report is produced. All the output summaries are total usage except number of times executed, CPU minutes, and real minutes, which are split into prime and non-prime.

-c Sort by total CPU time, rather than total kcore-minutes.

-j Combine all commands invoked only once under "***other".

-n Sort by number of command invocations.

-s Any file names encountered hereafter are already in internal summary format.

-t Process all records as total accounting records. The default internal summary format splits each field into prime and non-prime-time parts. This option combines the prime and non-prime time parts into a single field that is the total of both, and provides upward compatibility with old style acctcms internal summary format records.

使用例

例 1 Using the acctcms command.

A typical sequence for performing daily command accounting and for maintaining a running total is:

```
example% acctcms filename ... > today
example% cp total previoustotal
example% acctcms -s today previoustotal > total
example% acctcms -a -s today
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), acct(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

注意事項

Unpredictable output results if -t is used on new style internal summary format files, or if it is not used with old style internal summary format files.

名前

acctcon, acctcon1, acctcon2 – connect-time accounting

形式

```
/usr/lib/acct/acctcon [-l lineuse] [-o reboot]
/usr/lib/acct/acctcon1 [-p] [-t] [-l lineuse] [-o reboot]
/usr/lib/acct/acctcon2
```

機能説明

acctcon converts a sequence of login/logoff records to total accounting records (see the tacct format in acct.h(3HEAD)). The login/logoff records are read from standard input. The file /var/adm/wtmpx is usually the source of the login/logoff records; however, because it might contain corrupted records or system date changes, it should first be fixed using wtmpfix. The fixed version of file /var/adm/wtmpx can then be redirected to acctcon. The tacct records are written to standard output.

acctcon is a combination of the programs acctcon1 and acctcon2. acctcon1 converts login/logoff records, taken from the fixed /var/adm/wtmpx file, to ASCII output. acctcon2 reads the ASCII records produced by acctcon1 and converts them to tacct records. acctcon1 can be used with the -l and -o options, described below, as well as with the -p and -t options.

オプション

-p
: Print input only, showing line name, login name, and time (in both numeric and date/time formats).

-t
: acctcon1 maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session still in progress. The -t flag causes it to use, instead, the last time found in its input, thus assuring reasonable and repeatable numbers for non-current files.

-l *lineuse*
: *lineuse* is created to contain a summary of line usage showing line name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logoffs. This file helps track line usage, identify bad lines, and find software and hardware oddities. Hangup, termination of login(1) and termination of the login shell each generate logoff records, so that the number of logoffs is often three to four times the number of sessions. See init(1M) and utmpx(4).

-o *reboot*
: reboot is filled with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.

使用例

例 1 Using the acctcon command.

The acctcon command is typically used as follows:

```
example% acctcon -l lineuse -o reboots < tmpwtmp > ctacct
```

The acctcon1 and acctcon2 commands are typically used as follows:

例 1 Using the acctcon command. (続き)

```
example% acctcon1 -l lineuse -o reboots < tmpwtmp | sort +1n +2 > ctmp
example% acctcon2 < ctmp > ctacct
```

ファイル

`/var/adm/wtmpx` History of user access and administration information

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

`acctcom(1)`, `login(1)`, `acct(1M)`, `acctcms(1M)`, `acctmerg(1M)`, `acctprc(1M)`, `acctsh(1M)`, `fwtmp(1M)`, `init(1M)`, `runacct(1M)`, `acct(2)`, `acct.h(3HEAD)`, `utmpx(4)`, `attributes(5)`

『Oracle Solaris の管理: 一般的なタスク』

注意事項

The line usage report is confused by date changes. Use `wtmpfix` (see `fwtmp(1M)`), with the `/var/adm/wtmpx` file as an argument, to correct this situation.

During a single invocation of any given command, the `acctcon`, `acctcon1`, and `acctcon2` commands can process a maximum of:

- 6000 distinct session
- 1000 distinct terminal lines
- 2000 distinct login names

If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.

名前

acctmerg – merge or add total accounting files

形式

```
/usr/lib/acct/acctmerg [-a] [-i] [-p] [-t] [-u] [-v]
     [filename] ...
```

機能説明

acctmerg reads its standard input and up to nine additional files, all in the tacct format (see acct.h(3HEAD)) or an ASCII version thereof. It merges these inputs by adding records whose keys (normally user ID and name) are identical, and expects the inputs to be sorted on those keys.

オプション

- -a　Produce output in ASCII version of tacct.
- -i　Produce input in ASCII version of tacct.
- -p　Print input with no processing.
- -t　Produce a single record that totals all input.
- -u　Summarize by user ID, rather than by user ID and name.
- -v　Produce output in verbose ASCII format, with more precise notation for floating-point numbers.

使用例

例 1　Using the acctmerg command.

The following sequence is useful for making "repairs" to any file kept in this format:

```
example% acctmerg   -v   <filename1  >filename2
```

Edit *filename2* as you want:

```
example% acctmerg   -i   <filename2  >filename1
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), acct(1M), acctcms(1M), acctcon(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

『Oracle Solaris の管理: 一般的なタスク』

名前

acctprc, acctprc1, acctprc2 – process accounting

形式

```
/usr/lib/acct/acctprc
/usr/lib/acct/acctprc1 [ctmp]
/usr/lib/acct/acctprc2
```

機能説明

acctprc reads the standard input and converts it to total accounting records (see the tacct record in acct.h(3HEAD)). acctprc divides CPU time into prime time and non-prime time and determines mean memory size (in memory segment units). acctprc then summarizes the tacct records, according to user IDs, and adds login names corresponding to the user IDs. The summarized records are then written to the standard output. acctprc1 reads input in the form described by acct.h(3HEAD), adds login names corresponding to user IDs, then writes for each process an ASCII line giving user ID, login name, prime CPU time (tics), non-prime CPU time (tics), and mean memory size (in memory segment units). If *ctmp* is given, it should contain a list of login sessions sorted by user ID and login name. If this file is not supplied, it obtains login names from the password file, just as acctprc does. The information in *ctmp* helps it distinguish between different login names that share the same user ID.

From the standard input, acctprc2 reads records in the form written by acctprc1, summarizes them according to user ID and name, then writes the sorted summaries to the standard output as total accounting records.

使用例

例 1 Examples of acctprc.

The acctprc command is typically used as shown below:

```
example% acctprc  < /var/adm/pacct  > ptacct
```

The acctprc1 and acctprc2s commands are typically used as shown below:

```
example% acctprc1  ctmp  </var/adm/pacct
example% acctprc2 > ptacct
```

ファイル

/etc/passwd system password file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctsh(1M), cron(1M), fwtmp(1M), runacct(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

注意事項

Although it is possible for acctprc1 to distinguish among login names that share user IDs for commands run from a command line, it is difficult for acctprc1 to make this distinction for commands invoked in other ways. A command run from cron(1M) is an example of where

acctprc1 might have difficulty. A more precise conversion can be done using the acctwtmp program in acct(1M). acctprc does not distinguish between users with identical user IDs.

A memory segment of the mean memory size is a unit of measure for the number of bytes in a logical memory segment on a particular processor.

During a single invocation of any given command, the acctprc, acctprc1, and acctprc2 commands can process a maximum of

- 6000 distinct sessions
- 1000 distinct terminal lines
- 2000 distinct login names

If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.

名前

acctsh, chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, shutacct, startup, turnacct – shell procedures for accounting

形式

```
/usr/lib/acct/chargefee login-name number

/usr/lib/acct/ckpacct [blocks]

/usr/lib/acct/dodisk [-o] [filename]...

/usr/lib/acct/lastlogin

/usr/lib/acct/monacct number

/usr/lib/acct/nulladm filename...

/usr/lib/acct/prctmp filename

/usr/lib/acct/prdaily [-c] [-l] [mmdd]

/usr/lib/acct/prtacct filename [’’ heading ’’]

/usr/lib/acct/shutacct [’’ reason ’’]

/usr/lib/acct/startup

/usr/lib/acct/turnacct on | off | switch
```

機能説明

chargefee Command

`chargefee` can be invoked to charge a *number* of units to *login-name*. A record is written to `/var/adm/fee`, to be merged with other accounting records by `runacct(1M)`.

ckpacct Command

`ckpacct` should be initiated using `cron(1M)` to periodically check the size of `/var/adm/pacct`. If the size exceeds *blocks*, `500` by default, `turnacct` will be invoked with argument `switch`. To avoid a conflict with `turnacct switch` execution in `runacct`, do not run `ckpacct` and `runacct` simultaneously. If the number of free disk blocks in the `/var` file system falls below `500`, `ckpacct` will automatically turn off the collection of process accounting records via the `off` argument to `turnacct`. When at least `500` blocks are restored, the accounting will be activated again on the next invocation of `ckpacct`. This feature is sensitive to the frequency at which `ckpacct` is executed, usually by the `cron(1M)` command.

dodisk Command

`dodisk` should be invoked by `cron`(1M) to perform the disk accounting functions.

lastlogin Command

`lastlogin` is invoked by `runacct(1M)` to update `/var/adm/acct/sum/loginlog`, which shows the last date on which each person logged in.

monacct Command

`monacct` should be invoked once each month or each accounting period. *number* indicates which month or period it is. If *number* is not given, it defaults to the current month (01–12). This default is useful if `monacct` is to executed using `cron(1M)` on the first day of each month. `monacct` creates summary files in `/var/adm/acct/fiscal` and restarts the summary files in `/var/adm/acct/sum`.

nulladm Command

`nulladm` creates *filename* with mode 664 and ensures that owner and group are `adm`. It is called by various accounting shell procedures.

prctmp Command

`prctmp` can be used to print the session record file (normally `/var/adm/acct/nite/ctmp` created by `acctcon1` (see `acctcon(1M)`).

prdaily Command

`prdaily` is invoked by `runacct(1M)` to format a report of the previous day's accounting data. The report resides in `/var/adm/acct/sum/rprt/mmdd` where *mmdd* is the month and day of the report. The current daily accounting reports may be printed by typing `prdaily`. Previous days' accounting reports can be printed by using the *mmdd* option and specifying the exact report date desired.

prtacct Command

`prtacct` can be used to format and print any total accounting (`tacct`)file.

shutacct Command

`shutacct` is invoked during a system shutdown to turn process accounting off and append a *reason* record to `/var/adm/wtmpx`.

startup Command

`startup` can be invoked when the system is brought to a multi-user state to turn process accounting on.

turnacct Command

`turnacct` is an interface to `accton` (see `acct(1M)`) to turn process accounting `on` or `off`. The `switch` argument moves the current `/var/adm/pacct` to the next free name in `/var/adm/pacct.`*incr* (where *incr* is a number starting with `0` and incrementing by one for each additional `pacct` file), then turns accounting back on again. This procedure is called by `ckpacct` and thus can be taken care of by the `cron(1M)` command and used to keep `pacct` to a reasonable size. `shutacct` uses `turnacct` to stop process accounting. `startup` uses `turnacct` to start process accounting.

オプション

The following options are supported:

`-c` This option prints a report of exceptional resource usage by command, and may be used on current day's accounting data only.

`-l` This option prints a report of exceptional usage by login id for the specified date. Previous daily reports are cleaned up and therefore inaccessible after each invocation of `monacct`.

`-o` This option uses `acctdusg` (see `acct(1M)`) to do a slower version of disk accounting by login directory. *filename*s specifies the one or more filesystem names where disk accounting will be done. If *filename*s are used, disk accounting will be done on these filesystems only. If the `-o` option is used, *filename*s should be mount points of mounted filesystems. If the `-o` option is omitted, *filename*s should be the special file names of mountable filesystems.

ファイル

`/etc/logadm.conf`	Configuration file for the `logadm(1M)` command
`/usr/lib/acct`	Holds all accounting commands listed in section 1M of this manual

/usr/lib/acct/ptecms.awk	Contains the limits for exceptional usage by command name
/usr/lib/acct/ptelus.awk	Contains the limits for exceptional usage by login ID
/var/adm/acct/fiscal	Fiscal reports directory
/var/adm/acct/nite	Working directory
/var/adm/acct/sum	Summary directory that contains information for monacct
/var/adm/acct/sum/loginlog	File updated by last login
/var/adm/fee	Accumulator for fees
/var/adm/pacct	Current file for per-process accounting
/var/adm/pacct*incr*	Used if pacct gets large and during execution of daily accounting procedure
/var/adm/wtmpx	History of user access and administration information

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), cron(1M), fwtmp(1M), logadm(1M), runacct(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

注意事項

See runacct(1M) for the main daily accounting shell script, which performs the accumulation of connect, process, fee, and disk accounting on a daily basis. It also creates summaries of command usage.

名前

acpihpd – ACPI ホットプラグデーモン

形式

/usr/platform/i86pc/lib/acpihpd

機能説明

ACPI ホットプラグデーモン acpihpd は、 x86 プラットフォームで実行されるデーモンプロセスです。デーモンは、サービス管理機能によって開始されます。sysevent d(1M) および cfgadm(1M) サブシステムと通信して、ホットプラグ対応のデバイスから ACPI イベントを処理します。

acpihpd 用のサービス FMRI は次のとおりです。

svc:/platform/i86pc/acpihpd:default

acpihpd は非公開インタフェースであることに注意してください。

オプション

acpihpd デーモンでは、どのオプションもサポートされていません。

エラー

acpihpd は syslog(3C) を使用して、ステータスおよびエラーメッセージを報告します。すべてのメッセージは、LOG_DAEMON 機能を使用して記録されます。

エラーメッセージは LOG_ERR および LOG_NOTICE 優先順位を使用して記録され、情報メッセージは LOG_DEBUG 優先順位を使用して記録されます。/etc/syslog.conf ファイルのデフォルトエントリでは、すべてのエラーメッセージが /var/adm/messages ログに記録されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/kernel/dynamic-reconfiguration/i86pc
インタフェースの安定性	非公開

関連項目

svcs(1), cfgadm(1M), inetadm(1M), svcadm(1M), sysevent d(1M), syslog(3C), syslog.conf(4), attributes(5), smf(5)

注意事項

acpihpd サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

svc:/platform/i86pc/acpihpd:default

名前

adbgen – generate adb script

形式

`/usr/lib/adb/adbgen [-m model] filename.adb . . .`

機能説明

`adbgen` makes it possible to write `adb(1)` scripts that do not contain hard-coded dependencies on structure member offsets. The input to `adbgen` is a file named *filename*`.adb` that contains header information, then a null line, then the name of a structure, and finally an `adb` script. `adbgen` only deals with one structure per file; all member names are assumed to be in this structure. The output of `adbgen` is an `adb` script in *filename*. `adbgen` operates by generating a C program which determines structure member offsets and sizes, which in turn generate the `adb` script.

The header lines, up to the null line, are copied verbatim into the generated C program. Typically, these are `#include` statements, which include the headers containing the relevant structure declarations.

The `adb` script part may contain any valid `adb` commands (see `adb(1)`), and may also contain `adbgen` requests, each enclosed in braces (`{ }`). Request types are:

- Print a structure member. The request form is `{`*member*`,` *format*`}`. *member* is a member name of the *structure* given earlier, and `format` is any valid `adb` format request or any of the `adbgen` format specifiers (such as `{POINTER}`) listed below. For example, to print the `p_pid` field of the *proc* structure as a decimal number, you would write `{p_pid,d}`.
- Print the appropriate `adb` format character for the given `adbgen` format specifier. This action takes the data model into consideration. The request form is `{`*format specifier*`}`. The valid `adbgen` format specifiers are:

`{POINTER}`	pointer value in hexadecimal
`{LONGDEC}`	long value in decimal
`{ULONGDEC}`	unsigned long value in decimal
`{ULONGHEX}`	unsigned long value in hexadecimal
`{LONGOCT}`	long value in octal
`{ULONGOCT}`	unsigned long value in octal

- Reference a structure member. The request form is `{*`*member*`,` *base*`}`. *member* is the member name whose value is desired, and *base* is an `adb` register name which contains the base address of the structure. For example, to get the `p_pid` field of the *proc* structure, you would get the *proc* structure address in an `adb` register, for example `<f`, and write `{*p_pid,<f}`.
- Tell `adbgen` that the offset is valid. The request form is `{OFFSETOK}`. This is useful after invoking another `adb` script which moves the `adb` *dot*.
- Get the size of the *structure*. The request form is `{SIZEOF}`. `adbgen` replaces this request with the size of the structure. This is useful in incrementing a pointer to step through an array of structures.

- Calculate an arbitrary C expression. The request form is `{EXPR,` *expression*`}`. `adbgen` replaces this request with the value of the expression. This is useful when more than one structure is involved in the script.
- Get the offset to the end of the structure. The request form is `{END}`. This is useful at the end of the structure to get `adb` to align the *dot* for printing the next structure member.

`adbgen` keeps track of the movement of the `adb` *dot* and generates `adb` code to move forward or backward as necessary before printing any structure member in a script. `adbgen`'s model of the behavior of `adb`'s *dot* is simple: it is assumed that the first line of the script is of the form *struct_address*/*adb text* and that subsequent lines are of the form +/*adb text*. The `adb` *dot* then moves in a sane fashion. `adbgen` does not check the script to ensure that these limitations are met. `adbgen` also checks the size of the structure member against the size of the `adb` format code and warns if they are not equal.

オプション

The following option is supported:

`-m` *model* Specifies the data type model to be used by `adbgen` for the macro. This affects the outcome of the {*format specifier*} requests described under `DESCRIPTION` and the offsets and sizes of data types. *model* can be `ilp32` or `lp64`. If the `-m` option is not given, the data type model defaults to `ilp32`.

オペランド

The following operand is supported:

filename`.adb` Input file that contains header information, followed by a null line, the name of the structure, and finally an `adb` script.

使用例

例 1 A sample adbgen file.

For an include file `x.h` which contained

```
struct x {
         char    *x_cp;
         char    x_c;
         int    x_i;
};
```

then , an `adbgen` file (call it `script.adb`) to print the file `x.h` would be:

```
#include "x.h"
x
./"x_cp"16t"x_c"8t"x_i"n{x_cp,{POINTER}}{x_c,C}{x_i,D}
```

After running `adbgen` as follows,

```
% /usr/lib/adb/adbgen  script.adb
```

the output file `script` contains:

```
./"x_cp"16t"x_c"8t"x_i"nXC3+D
```

例 1　A sample adbgen file.　　(続き)

For a macro generated for a 64-bit program using the `lp64` data model as follows,

```
% /usr/lib/adb/adbgen/ -m lp64  script.adb
```

the output file `script` would contain:

```
./"x_cp"16t"x_c"8t"x_i"nJC3+D
```

To invoke the script, type:

```
example% adb program
x$<script
```

ファイル

`/usr/platform/`*platform-name*`/lib/adb/*`
　platform-specific `adb` scripts for debugging the 32-bit kernel

`/usr/platform/`*platform-name*`/lib/adb/sparcv9/*`
　platform-specific `adb` scripts for debugging the 64-bit SPARC V9 kernel

`/usr/lib/adb/*`
　`adb` scripts for debugging the 32-bit kernel

`/usr/lib/adb/sparcv9/*`
　`adb` scripts for debugging the 64-bit SPARC V9 kernel

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/debug/mdb

関連項目

`adb(1)`, `kmdb(1)`, `uname(1)`, `attributes(5)`

診断

Warnings are given about structure member sizes not equal to `adb` format items and about badly formatted requests. The C compiler complains if a structure member that does not exist is referenced. It also complains about an ampersand before array names; these complaints may be ignored.

注意事項

platform-name can be found using the `-i` option of `uname(1)`.

使用上の留意点

`adb` syntax is ugly; there should be a higher level interface for generating scripts.

Structure members which are bit fields cannot be handled because C will not give the address of a bit field. The address is needed to determine the offset.

名前

add_allocatable – 割り当てデータベースへのエントリの追加

形式

```
/usr/sbin/add_allocatable [-f] [-s] [-d] -n name -t type -l device-list
    [-a authorization] [-c clean] [-o key=value]
```

機能説明

add_allocatable は、ユーザーが割り当て可能なデバイス用の新規エントリを作成します。このデバイスは、デバイス割り当てメカニズムにより管理されます。add_allocatable を使用して、この種のデバイスの既存エントリを更新することもできます。

add_allocatable を使用して、デバイス割り当てメカニズムによりラベル範囲が管理される、プリンタなどの割り当て不可能なデバイスのエントリを作成および更新することもできます。

add_allocatable をシェルスクリプト (ドライバパッケージのインストールスクリプトなど) 内で実行して、新規デバイスの設定に関する管理作業を自動化できます。

割り当て可能なデバイスの名前とタイプ、属性、およびデバイスパスについては、list_devices(1) を参照してください。

オプション

-f	指定された情報を使って、既存のエントリを強制的に更新します。このオプションを指定しない場合、指定したデバイス名を持つエントリが既に存在すると、add_allocatable はエラーを出力して終了します。
-s	サイレントモードを有効にします。add_allocatable は、エラーや警告メッセージを一切出力しません。
-d	このオプションが指定されている場合、add_allocatable は、-t で指定されたデバイスタイプのシステム指定のデフォルト属性を更新します。
-n *name*	*name* で指定されたデバイスのエントリを追加または更新します。
-t *type*	*type* で指定されたタイプのデバイスエントリを追加または更新します。
-l *device-list*	-n で指定されたデバイスへのデバイスパスを追加または更新します。*device-list* 内の複数のパスは、スペースで区切る必要があります。また、リストは引用符で囲む必要があります。
-a *authorization*	-n で指定されたデバイス、または -t で指定されたタイプのデバイスに関連付けられている承認を追加または更新します。複数の承認を指定する場合は、リストをコンマで区切り、引用符で囲む必要があります。デバイスが割り当て不可能な場合、*authorization* をアスタリスク (*) を使って指定し、引用符で囲む必要があります。すべてのユーザーがデバイスを割り当て

可能である場合は、*authorization* をアットマーク (@) を使って指定し、引用符で囲む必要があります。デフォルトの承認は '@' です。

-c *clean*

-n で指定されたデバイス、または -t で指定されたデバイスタイプに対して device_clean(5) プログラム *clean* を使用するように指定します。デフォルトの clean プログラムは /bin/true です。

-o *key= value*

-n で指定されたデバイス、または -t で指定されたデバイスタイプに対して、コロンで区切られた *key=value* ペアの文字列を受け入れます。現在、システムにより解釈されるキーを次に示します。

minlabel　デバイスを使用可能な最下位のラベル。

maxlabel　デバイスを使用可能な最上位のラベル。

class　デバイスの論理グループを指定します。たとえば、すべての Sun Ray デバイスの、すべてのデバイスタイプは論理グループです。class キーワードにはデフォルト値はありません。

xdpy　X セッションの表示名を指定します。このキーワードは、X セッションに関連するデバイスを特定するために使用します。xdpy キーワードにはデフォルト値はありません。

終了ステータス

成功した場合、add_allocate は終了ステータス 0 (真) を返します。エラーが発生した場合、add_allocate は、0 以外の終了ステータスを返します。終了コードとその意味を次に示します。

1　呼び出し時の構文エラー

2　原因不明のシステムエラー

3　指定されたデバイスのエントリは既に存在します。このエラーは、-f オプションが指定されていない場合にのみ発生します。

4　アクセス権が与えられていません。ユーザーは、DAC または MAC アクセス記録の更新を保持していません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

呼び出しは不確実です。オプションは不確実です。出力は非インタフェースです。

関連項目

allocate(1), deallocate(1), list_devices(1), remove_allocatable(1M), attributes(5), device_clean(5)

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

名前

addbadsec – map out defective disk blocks

形式

`addbadsec [-p] [-a` *blkno* `[`*blkno*`]...] [-f` *filename*`]` *raw_device*

機能説明

`addbadsec` is used by the system administrator to map out bad disk blocks. Normally, these blocks are identified during surface analysis, but occasionally the disk subsystem reports unrecoverable data errors indicating a bad block. A block number reported in this way can be fed directly into `addbadsec`, and the block will be remapped. `addbadsec` will first attempt hardware remapping. This is supported on SCSI drives and takes place at the disk hardware level. If the target is an IDE drive, then software remapping is used. In order for software remapping to succeed, the partition must contain an alternate slice and there must be room in this slice to perform the mapping.

It should be understood that bad blocks lead to data loss. Remapping a defective block does not repair a damaged file. If a bad block occurs to a disk-resident file system structure such as a superblock, the entire slice might have to be recovered from a backup.

オプション

The following options are supported:

`-a` Adds the specified blocks to the hardware or software map. If more than one block number is specified, the entire list should be quoted and block numbers should be separated by white space.

`-f` Adds the specified blocks to the hardware or software map. The bad blocks are listed, one per line, in the specified file.

`-p` Causes `addbadsec` to print the current software map. The output shows the defective block and the assigned alternate. This option cannot be used to print the hardware map.

オペランド

The following operand is supported:

raw_device The address of the disk drive (see `FILES`).

ファイル

The raw device should be `/dev/rdsk/c?[t?]d?p0`. See `disks(1M)` for an explanation of SCSI and IDE device naming conventions.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86
Availability	system/core-os

関連項目

`disks(1M)`, `diskscan(1M)`, `fdisk(1M)`, `fmthard(1M)`, `format(1M)`, `attributes(5)`

注意事項

The format(1M) utility is available to format, label, analyze, and repair SCSI disks. This utility is included with the addbadsec, diskscan(1M), fdisk(1M), and fmthard(1M) commands available for x86. To format an IDE disk, use the DOS "format" utility; however, to label, analyze, or repair IDE disks on x86 systems, use the Solaris format(1M) utility.

名前

add_drv – システムへの新しいデバイスドライバの追加

形式

```
add_drv [-b basedir] [-c class_name]
    [-i 'identify_name...'] [-m 'permission','...']
    [-p 'policy'] [-P privilege] [-n] [-f] [-u] [-v] device_driver
```

機能説明

add_drv コマンドは、新たにインストールされたデバイスドライバについてシステムに通知するために使われます。

システム上の各デバイスには名前が関連付けられています。この名前は、デバイスの name プロパティーによって表されます。同様に、デバイスにはドライバ名のリストが関連付けられていることもあります。このリストは、デバイスの compatible プロパティーによって表されます。

システムは、各デバイスの name プロパティーと compatible プロパティー (存在する場合) の内容を調べることで、追加されるドライバによってどのデバイスが管理されるのかを判断します。name プロパティーの値が追加されるドライバと一致しない場合は、compatible プロパティーの各エントリが順番に試され、一致するエントリが見つかるか、compatible プロパティーのエントリがなくなるまで行われます。

場合によっては、新しいドライバを追加する際に再起動 (boot-r) が必要になることがあります。「注意事項」の節を参照してください。

エイリアスに数字が含まれている場合は、エイリアスを二重引用符で囲む必要があることがあります。「使用例」を参照してください。

/etc/minor_perm ファイル

add_drv および update_drv(1M) は、/etc/minor_perm ファイルを読み取ってアクセス権の情報を取得します。指定されたアクセス権は、ドライバにバインドされたデバイスが接続されるときに作成される、一致するマイナーノードに適用されます。マイナーノードのアクセス権は、chmod(1) によって手動で変更できます。このようなノードの場合、指定されたアクセス権が適用されて、add_drv または update_drv(1M) によって指定されたデフォルトのアクセス権が置き換えられます。

/etc/minor_perm ファイルの形式は次のとおりです。

name:*minor_name permissions owner group*

minor_name にはマイナーノードの実際の名前を指定することも、いくつかのマイナーノードを表すシェルのメタキャラクタを含めることもできます (sh(1) を参照)。

例:

```
sd:* 0640 root sys
zs:[a-z],cu 0600 uucp uucp
mm:kmem 0640 root bin
```

最初の行では、sd ノードによってエクスポートされるすべてのデバイスに、所有者 root、グループ sys を持つ 0640 アクセス権が設定されます。2 番目の行では、zs ドライバによってエクスポートされる a,cu や z,cu などのデバイスに、所有者

uucp、グループ uucp を持つ 0600 アクセス権が設定されます。3 番目の行では、mm ドライバによってエクスポートされる kmem デバイスに、所有者 root、グループ bin を持つ 0640 アクセス権が設定されます。

postinstall スクリプトからの add_drv の実行

パッケージの postinstall スクリプトのコンテキスト内から add_drv を実行するときは、そのパッケージがシステムイメージに追加されるのか、実行中のシステムに追加されるのかについて考慮する必要があります。パッケージをシステムイメージにインストールする場合、BASEDIR 変数はイメージのベースディレクトリを指します。この状況では、-b $BASEDIR を指定して add_drv を起動するようにしてください。これにより、add_drv はイメージのシステムファイルの更新だけを行います。ドライバを操作可能にするためには、システムまたはクライアントをリブートする必要があります。

パッケージが実行中のシステムそのものにインストールされる場合、前述の場合と同様に、システムファイルを更新する必要があります。ただし、リブートを行わなくても新しいドライバが存在することを実行中のカーネルに通知することができます。このためには、postinstall スクリプトが -b オプションなしで add_drv を起動する必要があります。したがって、add_drv を起動する postinstall スクリプトを次のように記述してください。

```
if [ "${BASEDIR:=/}" = "/" ]
then
        ADD_DRV="add_drv"
else
        ADD_DRV="add_drv -b ${BASEDIR}"
fi
$ADD_DRV [<options>] <driver>
```

あるいは、次のように記述してください。

```
if [ "${BASEDIR:=/}" != "/" ]
then
         BASEDIR_OPT="-b $BASEDIR"
fi
         add_drv $BASEDIR_OPT [<options>] <driver>
```

-b オプションについては、このあと説明します。

オプション

-b *basedir*

ドライバを、add_drv を実行しているシステムにインストールするのではなく、*basedir* というルートディレクトリを持つシステムにインストールします。このオプションは通常、pkgadd コマンドを実行しているシステムにパッケージがインストールされない場合に、パッケージのインストール後処理スクリプトで使われます。*basedir* をルートディレクトリとして使用するシステムは、リブートを行なってドライバのインストールを完了する必要があります。

注 – -b オプションを使って、非大域ゾーンのルートファイルシステムを参照しないでください。この操作によって、大域ゾーンのファイルシステムを損傷した

り、大域ゾーンのセキュリティーを損ねたり、非大域ゾーンのファイルシステムを損傷したりする可能性があります。zones(5) を参照してください。

-c *class_name*
システムに追加されるドライバによってクラス *class_name* がエクスポートされます。

-f
通常、システムへのドライバの設定を完了するために再起動 (boot-r) が必要な場合、add_drv はドライバを追加しません。強制 (force) フラグは、再起動 (boot-r) が必要な場合でも、add_drv に強制的にドライバを追加させます。-v フラグを参照してください。

-i '*identify_name*'
ドライバ *device_driver* のエイリアスをスペースで区切ったリスト。

-m '*permission*'
device_driver のためにシステムが作成したデバイスノードのファイルシステムアクセス権を指定します。

-n
device_driver のロードや接続を試みず、*device_driver* のシステム構成ファイルの変更だけを行います。

-p '*policy*'
追加のデバイスセキュリティーポリシーを指定します。

デバイスセキュリティーポリシーは、スペースで区切られたいくつかのトークンから構成されます。

{*minorspec* {token=value}+}+

minorspec は、マイナーデバイスの単純なワイルドカードパターンです。1 つの * はすべてのマイナーデバイスに適合します。パターンには、* を 1 つしか指定できません。

パターンは、次の順序で照合されます。

- ワイルドカードのないエントリ
- ワイルドカードのあるエントリ (長いワイルドカードから先に照合される)

定義されるトークンは、read_priv_set と write_priv_set です。read_priv_set は、読み取り用にデバイスを開くときに呼び出し側プロセスの実効セットで表明される必要がある特権を定義します。write_priv_set は、書き込み用にデバイスを開くときに呼び出し側プロセスの実効セットで表明される必要がある特権を定義します。privileges(5) を参照してください。

マイナーデバイス仕様 (minorspec) を指定しない場合は、* として解釈されます。

-P *'privilege'*
ドライバによって使用される追加の特権をコンマで区切って指定します。デバイスのポリシーに含まれる特定の特権を使用することもできます。

-u
ドライバをシステムに追加し、あとで devfsadm(1M) -u を使用して構成するためにアクティブではない状態のままにします。-u の動作が -n と異なるのは、-n はシステムファイルを更新するだけで、ドライバを接続するためにはリブートが必要になる点です。-u を指定して追加されたドライバは、devfsadm -u を実行することにより、リブートせずに接続できます。ドライバライターは、ドライバのこの動作について検証するようにしてください。その他の考慮事項については、「注意事項」を参照してください。-u オプションは -n または -b と同時に指定できません。

-v
冗長フラグを指定すると、add_drv はシステムへのドライバの設定の成功または失敗に関する追加情報を提供します。「使用例」の節を参照してください。

使用例

例 1　SUNW Example ドライバをシステムに追加する

次の例では、SUNW,alias というエイリアス名を使って、SUNW,example ドライバを 32 ビットシステムに追加します。ドライバは /usr/kernel/drv にすでにコピーされているものとします。

```
example# add_drv -m '* 0666 bin bin','a 0644 root sys' \
      -p 'a write_priv_set=sys_config  * write_priv_set=none' \
      -i 'SUNW,alias' SUNW,example
```

SUNW,example ドライバのためにシステムが作成する各マイナーノードは、アクセス権 0666 が与えられ、グループ bin のユーザー bin によって所有されます。ただし、マイナーデバイス a は例外であり、こちらはグループ sys のユーザー root によって所有され、アクセス権 0644 が与えられます。指定されたデバイスポリシーでは、すべてのマイナーモードを開くために追加の特権は必要ありませんが、マイナーデバイス a は例外であり、こちらは書き込み用にデバイスを開くときに sys_config 特権が必要です。

例 2　ドライバをクライアント /export/root/sun1 に追加する

次の例では、ドライバをクライアント /export/root/sun1 に追加します。クライアントマシン sun1 がリブートされると、ドライバがインストールされてロードされます。2 番目の例では、最初の例と同じ結果が得られますが、変更がディスクレスクライアントsun1 上に置かれ、ドライバをインストールするためにはクライアントをリブートする必要がある点が異なります。

```
example# add_drv -m '* 0666 bin bin','a 0644 root sys' \
        -i 'SUNW,alias' -b /export/root/sun1 \
    SUNW,example
```

例 2　ドライバをクライアント /export/root/sun1 に追加する　　*(続き)*

前述の -b オプションの説明に含まれる注意事項 (このオプションを Solaris ゾーン機能とともに使用する場合の注意事項が示されている) を参照してください。

例 3　既存のドライバによってすでに管理されているデバイスに対してドライバを追加する

次の例は、既存のドライバによってすでに管理されているデバイスに対して新しいドライバが追加される場合を示しています。デバイスがドライバ dumb_framebuffer によって現在管理されているとします。このデバイスの name および compatible プロパティーは、次のとおりです。

```
name="display"
compatible="whizzy_framebuffer", "dumb_framebuffer"
```

whizzy_framebuffer ドライバを追加するために add_drv を使用した場合は、次のような結果になります。

```
example# add_drv whizzy_framebuffer
Error: Could not install driver (whizzy_framebuffer)
Device managed by another driver.
```

-v フラグを指定した場合は、次のような結果になります。

```
example# add_drv -v whizzy_framebuffer
Error: Could not install driver (whizzy_framebuffer)
Device managed by another driver.
Driver installation failed because the following
entries in /devices would be affected:

        /devices/iommu@f,e0000000/sbus@f,e0001000/display[:*]
        (Device currently managed by driver "dumb_framebuffer")

The following entries in /dev would be affected:

        /dev/fbs/dumb_framebuffer0
```

-v および -f フラグを指定した場合は、ドライバが追加されて、次のような結果になります。

```
example# add_drv -vf whizzy_framebuffer
A reconfiguration boot must be performed to complete the
installation of this driver.

The following entries in /devices will be affected:

        /devices/iommu@f,e0000000/sbus@f,e0001000/display[:*]
        (Device currently managed by driver "dumb_framebuffer"

The following entries in /dev will be affected:
```

例 3 既存のドライバによってすでに管理されているデバイスに対してドライバを追加する (続き)

```
/dev/fbs/dumb_framebuffer0
```

前述の例は、現在のところ、汎用デバイス名をエクスポートするデバイスにのみ当てはまります。

例 4 ドライバエイリアスを指定する際に二重引用符を使用する

次の例では、数字が含まれるドライバエイリアスを指定する際に二重引用符を使用しています。

```
example# add_drv -i '"pci10c5,25"' smc
```

終了ステータス

add_drv は、成功すると 0 を返し、失敗すると 1 を返します。

ファイル

/kernel/drv
: 32 ビットブートデバイスドライバ

/kernel/drv/sparcv9
: 64 ビット SPARC ブートデバイスドライバ

/kernel/drv/amd64
: 64 ビット x86 ブートデバイスドライバ

/usr/kernel/drv
: 潜在的にプラットフォーム間で共有できる他の 32 ビットドライバ

/usr/kernel/drv/sparcv9
: 潜在的にプラットフォーム間で共有できる他の 64 ビット SPARC ドライバ

/usr/kernel/drv/amd64
: 潜在的にプラットフォーム間で共有できる他の 64 ビット x86 ドライバ

/platform/‘uname -i‘/kernel/drv
: 32 ビットプラットフォーム依存のドライバ

/platform/‘uname -i‘/kernel/drv/sparcv9
: 64 ビット SPARC プラットフォーム依存のドライバ

/platform/‘uname -i‘/kernel/drv/amd64
: 64 ビット x86 プラットフォーム依存のドライバ

/etc/driver_aliases
: ドライバエイリアスファイル

/etc/driver_classes
: ドライバクラスファイル

/etc/minor_perm
: マイナーノードのアクセス権

/etc/name_to_major
　メジャー番号のバインディング

/etc/security/device_policy
　デバイスポリシー

/etc/security/extra_privs
　デバイス特権

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

boot(1M), chmod(1), devfsadm(1M), kernel(1M), modinfo(1M), rem_drv(1M), update_drv(1M), driver.conf(4), system(4), attributes(5), privileges(5), devfs(7FS), ddi_create_minor_node(9F)

『Writing Device Drivers』

注意事項

別のドライバによってすでに管理されているデバイスに対してドライバを追加することが可能です。その場合、追加されるドライバは、デバイスの compatible リストの現在のドライバの前に表示されます。このような場合は、再起動 (boot-r) が必要です (boot(1M) および kernel(1M) を参照)。再起動 (boot-r) のあとは、/dev 内のデバイスリンクと、これらのファイルへの参照が無効になる場合があります (-v フラグを参照)。ドライバのインストールを完了するために再起動 (boot-r) が必要になる場合、-f オプションを指定しないかぎり、add_drv は失敗します。「使用例」の節の「例 3」を参照してください。

デバイスポリシーの導入によって、一部のドライバのマイナーアクセス権が変更されたり、デバイスポリシーがインストールされたりします。一般的なネットワークドライバには、次のデバイスポリシーを使用するようにしてください。

```
add_drv -p 'read_priv_set=net_rawaccess\
   write_priv_set=net_rawaccess' -m '* 666 root sys'\
   mynet
```

このマニュアルは、API を構成するものではありません。/etc/minor_perm、/etc/name_to_major、/etc/driver_classes、および /devices は、将来のリリースで削除されたり、内容や解釈が異なっていたりすることがあります。ここに記載された通知は、この通知のない他のドキュメントが API を構成していることを意味するものではありません。

/etc/minor_perm は、add_drv(1M)、rem_drv(1M)、または update_drv(1M) によってのみ更新できます。

現在のバージョンの add_drv は、コマンド行から使用する場合、別名を指定するための二重引用符の使用は省略できます。ただし、パッケージスクリプトから add_drv を使用する場合は、二重引用符を引き続き使用して別名を指定する必要があります。

いくつかのドライバは、システムに直接追加および構成せずに、システムブートとしてのみ構成してください。この制限の理由には、ブート初期の構成にドライバの依存性があるため、またはドライバの追加と同時にインストールまたは更新されるいくつかのカーネルコンポーネントに依存性があるためなどがあります。そのようなドライバは、システムのリブート時のみにドライバがロードおよび構成されるように、必ず -n フラグを指定してシステムに追加する必要があります。これにより、ドライバを適切に構成できる環境が保証されます。

使用上の留意点

旧バージョンの add_drv は、*device_driver* のパス名を受け入れていました。この機能はもうサポートされていないため、結果的にエラーが発生します。

名前

arp – address resolution display and control

形式

```
arp hostname

arp -a [-n]

arp -d hostname

arp -f filename

arp -s hostname ether_address [temp] [pub] [trail]
     [permanent]
```

機能説明

The `arp` program displays and modifies the Internet-to-MAC address translation tables used by the address resolution protocol (see `arp(7P)`).

With no flags, the program displays the current ARP entry for *hostname*. The host may be specified by name or by number, using Internet dot notation.

Options that modify the ARP translation tables (`-d`, `-f`, and `-s`) can be used only when the invoked command is granted the `sys_net_config` privilege. See `privileges(5)`.

オプション

`-a` Display all of the current ARP entries. The definition for the flags in the table are:

- `d` Unverified; this is a local IP address that is currently undergoing Duplicate Address Detection. ARP will not respond to requests for this address until Duplicate Address Detection completes.
- `o` Old; this entry is aging away. If IP requests it again, a new ARP query will be generated. This state is used for detecting peer address changes.
- `y` Delayed; periodic address defense and conflict detection was unable to send a packet due to internal network use limits for non-traffic-related messages (100 packets per hour per interface). This occurs only on interfaces with very large numbers of aliases.
- `A` Authority; this machine is authoritative for this IP address. ARP will not accept updates from other machines for this entry.
- `L` Local; this is a local IP address configured on one of the machine's logical interfaces. ARP will defend this address if another node attempts to claim it.
- `M` Mapping; only used for the multicast entry for `224.0.0.0`
- `P` Publish; includes IP address for the machine and the addresses that have explicitly been added by the `-s` option. ARP will respond to ARP requests for this address.
- `S` Static; entry cannot be changed by learned information. This indicates that the `permanent` flag was used when creating the entry.
- `U` Unresolved; waiting for ARP response.

You can use the `-n` option with the `-a` option to disable the automatic numeric IP address-to-name translation. Use `arp -an` or `arp -na` to display numeric IP addresses. The `arp -a` option is equivalent to:

```
# netstat -p -f inet
```

...and `-an` and `-na` are equivalent to:

```
# netstat -pn -f inet
```

`-d` Delete an entry for the host called *hostname*.

Note that ARP entries for IPMP (IP Network Multipathing) data and test addresses are managed by the kernel and thus cannot be deleted.

`-f` Read the file named *filename* and set multiple entries in the ARP tables. Entries in the file should be of the form:

hostname MACaddress `[temp] [pub] [trail] [permanent]`

See the `-s` option for argument definitions.

`-s` Create an ARP entry for the host called *hostname* with the MAC address *MACaddress*. For example, an Ethernet address is given as six hexadecimal bytes separated by colons. The entry will not be subject to deletion by aging unless the word `temp` is specified in the command. If the word `pub` is specified, the entry will be published, which means that this system will respond to ARP requests for *hostname* even though the *hostname* is not its own. The word `permanent` indicates that the system will not accept MAC address changes for *hostname* from the network.

Solaris does not implement trailer encapsulation, and the word `trail` is accepted on entries for compatibility only.

`arp -s` can be used for a limited form of proxy ARP when a host on one of the directly attached networks is not physically present on a subnet. Another machine can then be configured to respond to ARP requests using `arp -s`. This is useful in certain SLIP configurations.

Non-temporary proxy ARP entries for an IPMP (IP Network Multipathing) group are automatically managed by the kernel. Specifically, if the hardware address in an entry matches the hardware address of an IP interface in an IPMP group, and the IP address is not local to the system, this will be regarded as an IPMP proxy ARP entry. This entry will have its hardware address automatically adjusted in order to keep the IP address reachable so long as the IPMP group has not entirely failed.

ARP entries must be consistent across an IPMP group. Therefore, ARP entries cannot be associated with individual underlying IP interfaces in an IPMP group, and must instead be associated with the corresponding IPMP IP interface.

Note that ARP entries for IPMP data and test addresses are managed by the kernel and thus cannot be changed.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

ifconfig(1M), netstat(1M), attributes(5), privileges(5), arp(7P)

名前

asradm – Auto Service Request registration utility

形式

```
asradm list

asradm send [-n] [activate|audit|deactivate|fault|heartbeat|test]

asradm register -u username [-p password-file] [-e endpoint_URL]

asradm set-proxy [-i] | [-h host[:port]] [-u username] [-p password-file]

asradm unregister
```

機能説明

The `asradm` utility is used by a privileged system administrator to register hosts for enrollment in the Oracle Auto Service Request (ASR) for Oracle Sun systems. Using auto-case generation improves system availability and expedites the diagnostic process when specific hardware faults occur.

Once the system has been registered with a valid My Oracle Support (MOS) account the `smf(5)` service, `asr-notify`, will start to send HTTPS/XML telemetry either directly to the configured endpoint or through an optional HTTP proxy.

The `asradm` utility can also list the current registration state as seen from the registered server. This only confirms that messages are being sent to the Oracle ASR service and does not mean the system is actively being monitored. The user must log into their MOS account and activate the service for the system. An email should be sent to the user after registration that will describe the specific details needed to complete enrollment.

ASR messages can be sent manually using the `send` subcommand. The content of ASR messages can also be viewed without actually sending the messages as well. The generation of messages can be done even if the system has not been registered.

The system can also be unregistered from the ASR service, which will remove the system from being monitored and will disable all telemetry sent by the system.

オプション

The following options are supported:

`-e` *endpoint_URL*
: Sets the endpoint URL used for registration and all message telemetry. The default value is `transport.sun.com` sends all telemetry directly to Oracle service. A different URL can be used to support a local instance of an ASR Manager solution. The local ASR Manager can be used to aggregate telemetry from many hosts instances.

`-h` *host*[:*port*]
: Sets the HTTPS proxy host and optional port number to use for connecting to the internet. If a port is not specified, the default port value of `80` is used.

`-i`
: Sets the HTTPS connection to be a direct internet connection and thus not use any proxy host. This option will clear any previously set HTTPS proxy information.

`-n`
Do a dry run of sending an event, which displays the message data to stdout that would have been sent.

`-u` *username*
When used with the `register` subcommand, this specifies the MOS user name to be associated with registrations of products on this system. When used with the `set-proxy` subcommand this specifies the HTTPS proxy user name used for message transport.

`-p` *password-file*
This should be a single-line file containing a password value. It can be immediately removed after running this command. When used with the `register` subcommand, this specifies the password associated with the MOS user name. When used with the `set-proxy` subcommand, this option specifies the password associated with the HTTPS proxy used to connect to the internet. If the `-p` option is not entered and the `-u` option was set, the user will be prompted for the password.

サブコマンド

The `asradm` subcommands are described as follows.

`register`
Registers the system with MOS using the supplied authentication credentials. The MOS password will be used only for initial registration, to obtain a token used for all future telemetry and will not be stored anywhere on the system.

If the host system is behind a firewall, then the HTTP proxy settings must be set using the set-proxy command before registering the system.

Once registered, an ASR activation message will be sent that will request support for automated support call generation. Subsequently, an email will be sent to the registered user giving the status of the service request.

`list`
Shows the current authenticated MOS user name and the network connectivity information needed for HTTPS communication with MOS. If no registration has yet been done (by means of the `register` command), then the status of `Unregistered` is displayed.

`set-proxy`
Sets up the HTTPS connection information to be used for sending all ASR messages. Either a direct connection can be made or an HTTPS proxy can be defined.

`unregister`
Sends an ASR deactivation event and removes all configured registration information. No further telemetry will be sent and the Oracle ASR service will no longer generate any automated support calls.

`send`
Manually sends a specified ASR message to the Oracle ASR service or, with the dry run option (`-n`), display ASR messages.

使用例

例 1 Setup Internet Connection to Use an HTTPS Proxy

The following command will route all messages through an HTTPS proxy host `webproxy.example.com` on port 8080.

```
# asradm set-proxy -h webproxy.example.com:8080
```

例 2 Registering an MOS ID

The following command is used to interactively authenticate and register this system with the given MOS ID for use with ASR. Following this, you will be prompted for your support user name and password.

```
# asradm register
```

例 3 Authenticating Non-interactively

This is similar to `set–proxy` example, above. The difference is that the MOS user name and password are specified by means of the command line.

```
# asradm register -u joe.admin@example.com -p mypassword
```

例 4 Viewing Contents of Audit Message

The command below will display an audit message without sending the event. This will work even if the ASR service has not been registered.

```
# asradm send -n audit
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/fault-management/asr-notify
Interface Stability	Uncommitted

関連項目

`svcs(1)`, `asr-notify(1M)`, `svcadm(1M)`, `svccfg(1M)`, `syslogd(1M)`, `attributes(5)`, `smf(5)`

名前

asr-notify – auto service request notification daemon for Fault Management events

形式

```
/usr/lib/fm/notify/asr-notify [- ]
```

機能説明

`asr-notify` is a daemon that subscribes to Fault Management life cycle events and produces HTTPS/XML notifications based on a set of notification preferences that are stored in the SMF service configuration repository.

The messages will be sent to the Oracle Auto Service Request (ASR) service offering once the service has been registered using the `asradm(1M)` command or by setting SMF auto registration properties. If the auto registration properties are set, the service will attempt to register on either startup or refresh. If registration fails, the service will go into maintenance mode. If registration is successful, the SMF auto registration properties will be deleted.

`asr-notify` is managed by the service management facility, `smf(5)`, under the service FMRI:

```
svc:/system/fm/asr-notify:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

ASR notification preferences are set using `svccfg(1M)`.

Properties

The following service properties can be set:

`config/`*rootdir*
: This is an `astring` property that defaults to `/`. When set, the specified root directory will be used for all pathnames evaluated by `asr-notify`.

`config/`*http_timeout*
: This is an integer property that sets the number of seconds to wait for HTTP connections before generating an error.

`config/`*endpoint*
: This is an `astring` property that sets the endpoint URL used for registration and all message telemetry. The endpoint URL can be set to a local ASR Manager that will aggregate telemetry from many hosts instances. If not set, then all telemetry will be sent directly to the Oracle Auto Service Request service.

`autoreg/`*user*
: This is an `astring` property that is used to set the My Oracle Support user name used for auto registration.

`autoreg/`*password*
: This is an `astring` property that is used to set the My Oracle Support password used for auto registration.

`autoreg/`*proxy-host*
: This is an `astring` property that is used to set the HTTP proxy *hostname*:*port* and needs to be set only if an HTTP proxy is used to connect to the internet.

`autoreg/`*proxy-user*
: This is an `astring` property that is used to set the HTTP proxy user name and needs to be set only if the HTTP proxy requires credentials.

`autoreg/`*proxy-password*
: This is an `astring` property that is used to set the HTTP proxy password and needs to be set only if the HTTP proxy requires credentials.

Messages

The service will send the following types of messages:

- fault messages when an FMA fault is created
- daily heartbeat messages to indicate that the system is still active
- running and weekly audit messages to provide a full inventory of operational components

The contents of each message are an XML document that adheres to the ASR message XML schema. Each message starts with a common message header followed by event specific content.

ASR event header information:

```
XML Element     Description
------------    -----------
site-id         the serial number of the host sending
                the event
host-id         the hostname(1) of the system that the
                message is about
message-uuid    a unique id generated for each message
message-time    local time on the system that generated
                the message
system-id       the serial number of the system that the
                message is about
asset-id        the registered service tag for the system
product-id      the unique product identifier for the system
product-name    the name of the product for the system
event           element containing fault specific event
                information
```

The fault-specific content in the event will be contained within the primary-event-information XML element.

```
XML Element     Description
-----------     -----------
message-id      the knowledge article message ID
event-uuid      the FMA event UUID
event-time      the time that the FMA event was generated
severity        the severity of the FMA event
component       element containing list of suspects
summary         a short summary of the event
```

```
description     event description
payload         the raw FMA event contents
```

For each suspect FRU associated with a fault event, a hardware-component element will be included within the component.

```
XML Element              Description
-----------              -----------
name                     the component path name of the FRU
serial                   the FRU serial number
part                     the FRU part number
revision                 the FRU revision level
additional-information   a list of additional properties
                         containing values such as FRU
                         manufacturer and FRU model
```

Some faults might be associated with a software issue instead of a hardware issue. In this case, the fault event will contain a software-module element within the component list.

```
XML Element              Description
-----------              -----------
name                     the name or FMRI of the software
                         module in error
description              a description of the software
                         module error
```

Heartbeat events do not contain any additional data and audit events contain a list of all hardware-component, software-package, and software-module elements available to the system.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/fault-management/asr-notify
Interface Stability	Uncommitted

関連項目

svcs(1), asradm(1M), svcadm(1M), svccfg(1M), syslogd(1M), attributes(5), smf(5)

名前

atohexlabel – 人間が読めるラベルの内部テキスト形式への変換

形式

`/usr/sbin/atohexlabel [`*human-readable-sensitivity-label*`]`

`/usr/sbin/atohexlabel -c [`*human-readable-clearance*`]`

機能説明

`atohexlabel` は、人間が読むことのできるラベルを、公開オブジェクトに安全に格納可能な内部テキスト表現に変換します。オプションが指定されない場合、ラベルは機密ラベルと見なされます。

内部変換したあとで構文解析することで、同じ値に戻すことができます。多くの場合、内部形式は 16 進数で記述されます。変換されたラベルは、標準出力ファイルに書き込まれます。人間が読めるラベルを指定しない場合、標準入力ファイルからラベルが読み取られます。このコマンドは、内部データベース内に格納されているラベルを緊急に修復する場合に使用できます。

オプション

`-c` 人間の読めるラベルを認可上限として識別します。

終了ステータス

次の終了値が返されます。

`0` 成功の場合。

`1` エラーが発生しました。診断は標準エラーファイルに書き込まれます。

ファイル

`/etc/security/tsol/label_encodings`
ラベルエンコーディングファイルには、このシステムの定義済みラベルに対する格付け名、語句、制約、および値が含まれます。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

コマンド出力は、同一の `label_encodings` ファイルを使用する場合は「確実」です。コマンド呼び出しは、DIA MAC ポリシーを実装するシステムでは「確実」です。

関連項目

`hextoalabel(1M)`, `label_to_str(3TSOL)`, `str_to_label(3TSOL)`, `label_encodings(4)`, `attributes(5)`

『Trusted Extensions 構成と管理』

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合のみです。

このファイルは、米国国防情報局 (DIA) の必須アクセス制御 (MAC) ポリシーの一部です。このファイルは、Solaris Trusted Extensions ソフトウェアの将来のリリース向けに作成されるその他の MAC ポリシーには適用されない可能性があります。

名前

audit – control the behavior of the audit daemon

形式

```
audit -n | -s | -t | -v
```

機能説明

The `audit` command is the system administrator's interface to start, stop, and refresh the audit service `auditd(1M)`. Refreshing the audit service rereads the system and plugin configuration.

オプション

`-n` Notify the audit service `audit_binfile` plugin to close the current audit file and open a new audit file in the current audit directory.

`-s` Start (enable) the audit service if it is not running, or refresh the audit service, if it is currently running.

`-t` Terminate (disable) the audit service. The audit service will close out the active plugins, stop auditing and exit. Use `-s` to restart auditing.

`-v` Verify that at least one plugin is active. Verify the directory parameters of `audit_binfile(5)`.

診断

The `audit` command will exit with `0` upon success and a positive integer upon failure.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`auditconfig(1M)`, `auditd(1M)`, `attributes(5)`, `audit_binfile(5)`

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

注意事項

The `audit` command does not modify a process's preselection mask. Its functions are limited to performing control actions of the auditing subsystem. See `auditconfig(1M)` for configuration.

The `-s` option validates the audit plugin configuration. If it is not valid an error message is displayed and the audit service is not started or refreshed. The `-v` option may be used to validate the audit plugin configuration before using the `-s` option to start or refresh the audit service.

The `-s` option also checks state of the audit service. In case the audit service is found in the maintenance state (thus not able to be enabled or refreshed) the `audit` command returns with an appropriate message and exit code.

All options are valid in the global zone. Unless per-zone is enabled, only the `-v` option is valid in a local zone. See `auditconfig(1M)` for per-zone audit configuration.

名前 auditconfig – configure auditing

形式 `auditconfig` *subcommand*...

機能説明 `auditconfig` provides a command line interface to get and set kernel audit parameters.

Except for getting or setting the persistent audit service values, this functionality is available only if the Solaris Auditing feature has been enabled.

A zero (`0`) queue value indicates that the system default is in effect.

The setting of the `perzone` policy determines the scope of the audit setting controlled by `auditconfig`. If `perzone` is set, then the values reflect the local zone except as noted. Otherwise, the settings are for the entire system. Any restriction based on the `perzone` setting is noted for each option to which it applies.

A non-global zone administrator can set all audit policy options except `perzone` and `ahlt`. `perzone` and `ahlt` apply only to the global zone; setting these policies requires the privileges of a global zone administrator. `perzone` and `ahlt` are described under the `-setpolicy` option, below.

This command is available to administrators who have been granted the Audit Control Rights Profile.

オプション The following option is supported:

`-t`
Display or set the values on the running system in addition to the persistent values of the audit service.

This option is available only for the subcommands that list it below.

サブコマンド

`-aconf`
Set the configured non-attributable audit mask. For example:

```
# auditconfig -aconf
Configured non-attributable event mask.
```

`-audit` *event sorf retval string*
This command constructs an audit record for audit event *event* using the process's audit characteristics containing a text token *string*. The return token is constructed from the *sorf* (success/failure flag) and the *retval* (return value). The event is type `char*`, the *sorf* is 0/1 for success/failure, *retval* is an errno value, *string* is type `*char`. This command is useful for constructing an audit record with a shell script. An example of this option:

```
# auditconfig -audit AUE_ftpd 0 0 "test string"
#

audit record from audit trail:
    header,76,2,ftp access,,Fri Dec 08 08:44:02 2000, + 669 msec
```

```
subject,abc,root,other,root,other,104449,102336,235 197121 elbow
text,test string
return,success,0
```

-chkaconf
: Checks the configuration of the non-attributable events set in the kernel against the entries configured in the audit service (-setnaflags). If the active class mask of a kernel audit event does not match the configured class mask, a mismatch is reported.

-chkconf
: Check the configuration of kernel audit event to class mappings. If the runtime class mask of a kernel audit event does not match the configured class mask, a mismatch is reported.

-conf
: Configure kernel audit event to class mappings. Runtime class mappings are changed to match those in the audit event to class database file.

-getasid
: Prints the audit session ID of the current process. For example:

```
# auditconfig -getasid
audit session id = 102336
```

-getaudit
: Returns the audit characteristics of the current process.

```
# auditconfig -getaudit
audit id = abc(666)
process preselection mask = lo(0x1000,0x1000)
terminal id (maj,min,host) = 235,197121,elbow(172.146.89.77)
audit session id = 102336
```

-getauid
: Prints the audit ID of the current process. For example:

```
# auditconfig -getauid
audit id = abc(666)
```

-getcar
: Prints current active root location (anchored from root [or local zone root] at system boot). For example:

```
# auditconfig -getcar
current active root = /
```

-getclass *event*
: Display the preselection mask associated with the specified kernel audit event. *event* is the kernel event number or event name.

-getcond
: Display the kernel audit condition. The condition displayed is the literal string auditing meaning auditing is enabled and turned on (the kernel audit module is constructing and

queuing audit records, audit daemon is running); `noaudit`, meaning auditing is enabled but turned off (the kernel audit module is not constructing and queuing audit records, audit daemon is not running); `disabled`, meaning that the audit module has not been enabled (the module has been excluded in `system(4)`). See `auditd(1M)` for further information.

`-getestate` *event*
For the specified event (string or event number), print out classes *event* has been assigned. For example:

```
# auditconfig -getestate 20
audit class mask for event AUE_REBOOT(20) = 0x800
# auditconfig -getestate AUE_RENAME
audit class mask for event AUE_RENAME(42) = 0x30
```

`-getflags`
Display the user default audit preselection flags.

`-getkaudit`
Get audit characteristics of the current zone. For example:

```
# auditconfig -getkaudit
audit id = unknown(-2)
process preselection mask = lo,na(0x1400,0x1400)
terminal id (maj,min,host) = 0,0,(0.0.0.0)
audit session id = 0
```

If the audit policy `perzone` is not set, the terminal id is that of the global zone. Otherwise, it is the terminal id of the local zone.

`-getkmask`
Get non-attributable pre-selection mask for the current zone. For example:

```
# auditconfig -getkmask
audit flags for non-attributable events = lo,na(0x1400,0x1400)
```

If the audit policy `perzone` is not set, the kernel mask is that of the global zone. Otherwise, it is that of the local zone.

`-getnaflags`
Display the non-attributable audit flags.

`-getpinfo` *pid*
Display the audit ID, preselection mask, terminal ID, and audit session ID for the specified process.

`-getplugin` [*name*]
Display information about the plugin name. If *name* is not specified, display all plugins.

[-t] -getpolicy
: Display the kernel audit policy. The `ahlt` and `perzone` policies reflect the settings from the global zone. If `perzone` is set, all other policies reflect the local zone's settings. If `perzone` is not set, the policies are machine-wide.

-getcwd
: Prints current working directory (anchored from zone root at system boot). For example:

```
# cd /usr/tmp
# auditconfig -getcwd
current working directory = /var/tmp
```

[-t] -getqbufsz
: Get audit queue write buffer size. For example:

```
# auditconfig -getqbufsz
no configured audit queue size
audit queue buffer size (bytes) = 1024
```

[-t] -getqctrl
: Get audit queue write buffer size, audit queue `hiwater` mark, audit queue `lowater` mark, audit queue `prod` interval (ticks).

```
# auditconfig -getqctrl
no configured audit queue lowater mark
no configured audit queue hiwater mark
no configured audit queue size
no configured audit queue delay
audit queue hiwater mark (records) = 100
audit queue lowater mark (records) = 10
audit queue buffer size (bytes) = 1024
audit queue delay (ticks) = 20

# auditconfig -setqbufsz 8192
# auditconfig -t -setqbufsz 12288
# auditconfig -setqdelay 20
# auditconfig -t -setqdelay 25
# auditconfig -getqctrl
no configured audit queue lowater mark
no configured audit queue hiwater mark
configured audit queue buffer size (bytes) = 8192
configured audit queue delay (ticks) = 20
active audit queue hiwater mark (records) =     100
active audit queue lowater mark (records) =     10
active audit queue buffer size (bytes) = 12288
active audit queue delay (ticks) = 25
```

[-t] -getqdelay
: Get interval at which audit queue is prodded to start output. For example:

```
# auditconfig -getqdelay
no configured audit queue delay
audit queue delay (ticks) = 20
```

[-t] -getqhiwater
Get high water point in undelivered audit records when audit generation will block. For example:

```
# ./auditconfig -getqhiwater
no configured audit queue hiwater mark
audit queue hiwater mark (records) = 100
```

[-t] -getqlowater
Get low water point in undelivered audit records where blocked processes will resume. For example:

```
# auditconfig -getqlowater
no configured audit queue lowater mark
audit queue lowater mark (records) = 10
```

-getstat
Print current audit statistics information. For example:

```
# auditconfig -getstat
gen nona kern  aud  ctl  enq wrtn wblk rblk drop  tot  mem
910    1  725  184    0  910  910    0  231    0   88   48
```

See auditstat(1M) for a description of the headings in -getstat output.

-gettid
Print audit terminal ID for current process. For example:

```
# auditconfig -gettid
terminal id (maj,min,host) = 235,197121,elbow(172.146.89.77)
```

-lsevent
Display the currently configured (runtime) kernel and user level audit event information.

-lspolicy
Display the kernel audit policies with a description of each policy.

-setasid *session-ID* [*cmd*]
Execute shell or *cmd* with specified *session-ID*. For example:

```
# ./auditconfig -setasid 2000 /bin/ksh
#
# ./auditconfig -getpinfo 104485
audit id = abc(666)
process preselection mask = lo(0x1000,0x1000)
terminal id (maj,min,host) = 235,197121,elbow(172.146.89.77)
audit session id = 2000
```

`-setaudit` *audit-ID preselect_flags term-ID session-ID* [*cmd*]
Execute shell or *cmd* with the specified audit characteristics.

`-setauid` *audit-ID* [*cmd*]
Execute shell or *cmd* with the specified *audit–ID*.

`-setclass` *event audit_flag*[,*audit_flag* . . .]
Map the kernel event *event* to the classes specified by `audit_flag` list. *event* is an event number or name. An *audit_flag* is a character string representing an audit class. See `audit_flags(5)` for further information. If `perzone` is not set, this option is valid only in the global zone.

`-setflags` *audit_flags*
Set the default user audit preselection flags; see `audit_flags(5)`. The default preselection flags are combined with the user's specific audit flags to form the user's audit preselection mask.

`-setkaudit` *IP-address_type IP_address*
Set IP address of machine to specified values. *IP-address_type* is `ipv6` or `ipv4`.

If `perzone` is not set, this option is valid only in the global zone.

`-setkmask` *audit_flags*
Set non-attributable preselection flags of machine.

If `perzone` is not set, this option is valid only in the global zone.

`-setnaflags` *audit_flags*
Set the non-attributable audit flags; see `audit_flags(5)`. Non-attributable audit flags define which classes of events are to be audited when the action cannot be attributed to an authenticated user. Failed login is an example of an event that is non-attributable.

`-setpmask` *pid flags*
Set the preselection mask of the specified process. *flags* is the ASCII representation of the flags similar to that in `audit_flags(5)`.

If `perzone` is not set, this option is valid only in the global zone.

`-setplugin` *name* `active`|`inactive` [*attributes* [*qsize*]]
Configure the plugin name to be `active` or `inactive`. Optionally configure the attributes and number of unprocessed audit records to queue for the plugin. See the relevant audit plugin man pages and `auditd(1M)`.

[`-t`] `-setpolicy` [+|-]*policy_flag*[,*policy_flag* ...]
Set the kernel audit policy. A policy *policy_flag* is literal strings that denotes an audit policy. A prefix of + adds the policies specified to the current audit policies. A prefix of - removes the policies specified from the current audit policies. No policies can be set from a local zone unless the `perzone` policy is first set from the global zone. The following are the valid policy flag strings (`auditconfig -lspolicy` also lists the current valid audit policy flag strings):

`all`	Include all policies that apply to the current zone.
`ahlt`	Panic is called and the system dumps core if an asynchronous audit event occurs that cannot be delivered because the audit queue has reached the high-water mark or because there are insufficient resources to construct an audit record. By default, records are dropped and a count is kept of the number of dropped records.
`arge`	Include the `execv(2)` system call environment arguments to the audit record. This information is not included by default.
`argv`	Include the `execv(2)` system call parameter arguments to the audit record. This information is not included by default.
`cnt`	Do not suspend processes when audit resources are exhausted. Instead, drop audit records and keep a count of the number of records dropped. By default, process are suspended until audit resources become available.
`group`	Include the supplementary group token in audit records. By default, the group token is not included.
`none`	Include no policies. If used in other than the global zone, the `ahlt` and `perzone` policies are not changed.
`path`	Add secondary path tokens to audit record. These are typically the pathnames of dynamically linked shared libraries or command interpreters for shell scripts. By default, they are not included.
`perzone`	Maintain separate configuration, queues, and logs for each zone and execute a separate version of `auditd(1M)` for each zone.
`public`	Audit public files. By default, read-type operations are not audited for certain files which meet *public* characteristics: owned by root, readable by all, and not writable by all.
`trail`	Include the trailer token in every audit record. By default, the trailer token is not included.
`seq`	Include the sequence token as part of every audit record. By default, the sequence token is not included. The sequence token attaches a sequence number to every audit record.
`windata_down`	Include in an audit record any downgraded data moved between windows. This policy is available only if the system is configured with Trusted Extensions. By default, this information is not included.
`windata_up`	Include in an audit record any upgraded data moved between windows. This policy is available only if the system is configured with Trusted Extensions. By default, this information is not included.

`zonename` Include the `zonename` token as part of every audit record. By default, the `zonename` token is not included. The `zonename` token gives the name of the zone from which the audit record was generated.

[`-t`] `-setqbufsz` *buffer_size*
Set the audit queue write buffer size (bytes). Zero (`0`), indicates reset to no configured value.

[`-t`] `-setqctrl` *hiwater lowater bufsz interval*
Set the audit queue write buffer size (bytes), hiwater audit record count, lowater audit record count, and wakeup interval (ticks). Valid within a local zone only if `perzone` is set. Zero (`0`), indicates reset to no configured value.

[`-t`] `-setqdelay` *interval*
Set the audit queue wakeup interval (ticks). This determines the interval at which the kernel pokes the audit queue, to write audit records to the audit trail. Valid within a local zone only if `perzone` is set. Zero (`0`), indicates reset to no configured value.

[`-t`] `-setqhiwater` *hiwater*
Set the number of undelivered audit records in the audit queue at which audit record generation blocks. Valid within a local zone only if `perzone` is set. Zero (`0`), indicates reset to no configured value.

[`-t`] `-setqlowater` *lowater*
Set the number of undelivered audit records in the audit queue at which blocked auditing processes unblock. Valid within a local zone only if `perzone` is set. Zero (`0`), indicates reset to no configured value.

`-setsmask` *asid flags*
Set the preselection mask of all processes with the specified audit session ID. Valid within a local zone only if `perzone` is set.

`-setstat`
Reset audit statistics counters. Valid within a local zone only if `perzone` is set.

`-setumask` *username|auid flags*
Set the preselection mask of all processes with the specified username or audit ID. Valid within a local zone only if `perzone` is set.

使用例

例 1 Using auditconfig

The following is an example of an `auditconfig` command:

```
#
# map kernel audit event number 10 to the "fr" audit class
#
% auditconfig -setclass 10 fr

#
# turn on inclusion of exec arguments in exec audit records
#
```

例 1 Using auditconfig (続き)

```
% auditconfig -setpolicy +argv
```

例 2 Setting Only the Number of Unprocessed Audit Records

The following sequence of commands sets only the number of unprocessed audit records to queue for the `audit_binfile` plugin:

```
#
# see if audit_binfile is active
#
% auditconfig -getplugin audit_binfile

#
# set to queue 20 unprocessed audit records
#
% auditconfig -setplugin active "" 20
```

例 3 Resetting Queue Control Parameters

The following commands reset active and configured queue control parameters:

```
#
# reset configured audit queue control parameters
#
% auditconfig -setqctrl 0 0 0 0

#
# reset active audit queue control parameters
#
% auditconfig -t -setqctrl 0 0 0 0
```

終了ステータス

`0` Successful completion.

`1` An error occurred.

ファイル

`/etc/security/audit_event`	Stores event definitions used in the audit system.
`/etc/security/audit_class`	Stores class definitions used in the audit system.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

audit(1M), auditd(1M), auditstat(1M), praudit(1M), execv(2), audit_class(4), audit_event(4), system(4), attributes(5), audit_binfile(5), audit_flags(5), audit_remote(5), audit_syslog(5)

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

注意事項

If plugin output is selected using the -setplugin option, the behavior of the system with respect to the -setpolicy +cnt and the -setqhiwater options is modified slightly. If -setpolicy +cnt is set, data will continue to be sent to the selected plugin, even though output of the audit_binary plugin is stopped, pending the freeing of disk space. If -setpolicy –cnt is used, the blocking behavior is as described under SUBCOMMANDS, above. The queue high water mark value is used within auditd as the upper bound for its queue limits unless overridden by means of the qsize attribute, as described in the explanation of the -setplugin option, above.

The auditconfig options that modify or display process-based information are not affected by the perzone policy. Those that modify system audit data such as the terminal id and audit queue parameters are valid only in the global zone, unless the perzone policy is set. The display of a system audit reflects the local zone if perzone is set. Otherwise, it reflects the settings of the global zone.

The change to plugins (-setplugin) settings do not take effect (such as becoming active or inactive, or changing the active attribute or queue size values) until the audit service is refreshed. Use audit(1M) to refresh the audit service.

名前

auditd – audit daemon

形式

/usr/sbin/auditd

機能説明

The audit daemon, auditd, controls the generation and location of audit trail files and the generation of syslog messages based on its configuration (see auditconfig(1M)). If auditing is enabled, auditd reads its configuration to do the following:

- Configure audit policy.
- Configure the audit queue control parameters.
- Configure the event-to-class mappings.
- Set the default audit masks.
- Load one or more plugins.

  Solaris provides three plugins. audit_binfile(5) writes binary audit data to a file. audit_remote(5) sends binary audit data to an authenticated server with privacy and integrity protection. audit_syslog(5) sends text summaries of audit records to the syslog daemon.
- Read audit data from the kernel and pass that data to each of the active plugins.
- Execute the audit_warn(1M) script to warn of various conditions.

audit(1M) is used to control auditd. It can cause auditd to:

- Close the current audit file and open a new one.
- Start and refresh the service based on the current properties.
- Close the audit trail and terminate auditing.

auditconfig(1M) is used to configure auditd. It can configure the active and permanent:

- audit policy
- audit queue control parameters
- default audit masks
- which plugins are to be loaded
- plugin attributes

Audit Record Queue

The maximum number of records to queue for audit data sent to the plugin is specified by the qsize parameter specified for the plugin. If omitted, the current hiwater mark is used. See the -getqctrl option in auditconfig(1M). When this maximum is reached, auditd will either block processes or discard data, depending on the cnt audit policy as described in auditconfig(1M).

Auditing Conditions

The audit daemon and audit plugins invoke the program audit_warn(1M) under certain conditions. See audit_warn(1M) for more information.

ファイル

- etc/security/audit/audit_class
- etc/security/audit/audit_event

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

audit(1M), audit_warn(1M), auditconfig(1M), praudit(1M), audit_class(4), audit_class(4), audit_event(4), attributes(5), audit_binfile(5), audit_flags(5), audit_remote(5), audit_syslog(5), smf(5)

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

注意事項

auditd is loaded in the global zone at boot time if auditing is enabled.

If the audit policy perzone is set, auditd runs in each zone, starting automatically when the local zone boots. If a zone is running when the perzone policy is set, auditing must be started manually in local zones. It is not necessary to reboot the system or the local zone to start auditing in a local zone. auditd can be started with audit -s and will start automatically with future boots of the zone.

When auditd runs in a local zone, the configuration is taken from the local zone's smf(5) repository and the /etc/security directory's files: audit_class, user_attr, and audit_event.

Configuration changes do not affect audit sessions that are currently running, as the changes do not modify a process's preselection mask. To change the preselection mask on a running process, use the –setpmask option of the auditconfig command (see auditconfig(1M)). If the user logs out and logs back in, the new configuration changes will be reflected in the next audit session.

The audit service FMRI is svc:/system/auditd:default.

名前

auditrecord – display Solaris audit record formats

形式

```
/usr/sbin/auditrecord [-d] [ [-a] | [-e string] | [-c class] |
      [-i id] | [-p programname] | [-s systemcall] | [-h]]
```

機能説明

The `auditrecord` utility displays the event ID, audit class and selection mask, and record format for audit record event types defined in `audit_event(4)`. You can use `auditrecord` to generate a list of all audit record formats, or to select audit record formats based on event class, event name, generating program name, system call name, or event ID.

There are two output formats. The default format is intended for display in a terminal window; the optional HTML format is intended for viewing with a web browser.

Tokens contained in square brackets (`[ ]`) are optional and might not be present in every record.

オプション

The following options are supported:

`-a`
List all audit records.

`-c` *class*
List all audit records selected by *class*. *class* is one of the two-character class codes from the file `/etc/security/audit_class`.

`-d`
Debug mode. Display number of audit records that are defined in `audit_event`, the number of classes defined in `audit_class`, any mismatches between the two files, and report which defined events do not have format information available to `auditrecord`.

`-e` *string*
List all audit records for which the event ID label contains the string *string*. The match is case insensitive.

`-h`
Generate the output in HTML format.

`-i` *id*
List the audit records having the numeric event ID *id*.

`-p` *programname*
List all audit records generated by the program *programname*, for example, audit records generated by a user-space program.

`-s` *systemcall*
List all audit records generated by the system call *systemcall*, for example, audit records generated by a system call.

The `-p` and `-s` options are different names for the same thing and are mutually exclusive. The `-a` option is ignored if any of `-c`, `-e`, `-i`, `-p`, or `-s` are given. Combinations of `-c`, `-e`, `-i`, and either `-p` or `-s` are ANDed together.

使用例

例 1 Displaying an Audit Record with a Specified Event ID

The following example shows how to display the contents of a specified audit record.

```
% auditrecord -i 6152
  terminal login
  program     /usr/sbin/login      see login(1)
              /usr/dt/bin/dtlogin  See dtlogin
  event ID    6152                 AUE_login
  class       lo                   (0x00001000)
      header
      subject
      [text]                       error message
      return
```

例 2 Displaying an Audit Record with an Event ID Label that Contains a Specified String

The following example shows how to display the contents of a audit record with an event ID label that contains the string `login`.

```
# auditrecord -e login
terminal login
  program     /usr/sbin/login      see login(1)
              /usr/dt/bin/dtlogin  See dtlogin
  event ID    6152                 AUE_login
  class       lo                   (0x00001000)
      header
      subject
      [text]                       error message
      return

rlogin
  program     /usr/sbin/login      see login(1) - rlogin
  event ID    6155                 AUE_rlogin
  class       lo                   (0x00001000)
      header
      subject
      [text]                       error message
      return
```

終了ステータス

`0`
: Successful operation

`non-zero`
: Error

ファイル

`/etc/security/audit_class`
: Provides the list of valid classes and the associated audit mask.

`/etc/security/audit_event`
: Provides the numeric event ID, the literal event name, and the name of the associated system call or program.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Uncommitted

関連項目

`auditconfig(1M)`, `praudit(1M)`, `audit.log(4)`, `audit_class(4)`, `audit_event(4)`, `attributes(5)`

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

診断

If unable to read either of its input files or to write its output file, `auditrecord` shows the name of the file on which it failed and exits with a non-zero return.

If no options are provided, if an invalid option is provided, or if both `-s` and `-p` are provided, an error message is displayed and `auditrecord` displays a usage message then exits with a non-zero return.

注意事項

This command was formerly known as `bsmrecord`.

If `/etc/security/audit_event` has been modified to add user-defined audit events, `auditrecord` displays the record format as `undefined`.

The audit records displayed by `auditrecord` are the core of the record that can be produced. Various audit policies and optional tokens, such as those shown below, might also be present.

The following is a list of `praudit(1M)` token names with their descriptions.

`group`
: Present if the `group` audit policy is set.

`sensitivity label`
: Present when Trusted Extensions is enabled and represents the label of the subject or object with which it is associated. The `mandatory_label` token is noted in the basic audit record where a label is explicitly part of the record.

`sequence`
: Present when the `seq` audit policy is set.

`trailer`
: Present when the `trail` audit policy is set.

`zone`

The name of the zone generating the record when the `zonename` audit policy is set. The `zonename` token is noted in the basic audit record where a zone name is explicitly part of the record.

名前

auditreduce – merge and select audit records from audit trail files

形式

`auditreduce [options] [audit-trail-file]...`

機能説明

`auditreduce` allows you to select or merge records from audit trail files. Audit files can be from one or more machines.

The merge function merges together audit records from one or more input audit trail files into a single output file. The records in an audit trail file are assumed to be sorted in chronological order (oldest first) and this order is maintained by `auditreduce` in the output file.

Unless instructed otherwise, `auditreduce` will merge the entire audit trail, which consists of all the audit trail files in the directory structure *audit_root_dir/** (see `audit.log(4)` for details of the structure of the audit root). Unless specified with the `-R` or `-S` option, *audit_root_dir* defaults to `/var/audit`. By using the file selection options it is possible to select some subset of these files, or files from another directory, or files named explicitly on the command line.

The select function allows audit records to be selected on the basis of numerous criteria relating to the record's content (see `audit.log(4)` for details of record content). A record must meet all of the *record-selection-option* criteria to be selected.

Audit Trail Filename Format

Any audit trail file not named on the command line must conform to the audit trail filename format. Files produced by the audit system already have this format. Output file names produced by `auditreduce` are in this format. It is:

start-time`.` *end-time*`.` *suffix*

where *start-time* is the 14-character timestamp of when the file was opened, *end-time* is the 14-character timestamp of when the file was closed, and *suffix* is the name of the machine which generated the audit trail file, or some other meaningful suffix (for example, `all`, if the file contains a combined group of records from many machines). The *end-time* can be the literal string `not_terminated`, to indicate that the file is still being written to by the audit system. Timestamps are of the form *yyyymmddhhmmss* (year, month, day, hour, minute, second). The timestamps are in Coordinated Universal Time (UTC).

オプション

File Selection Options

The file selection options indicate which files are to be processed and certain types of special treatment.

`-A`

All of the records from the input files will be selected regardless of their timestamp. This option effectively disables the `-a`, `-b`, and `-d` options. This is useful in preventing the loss of records if the `-D` option is used to delete the input files after they are processed. Note, however, that if a record is *not* selected due to another option, then `-A` will not override that.

`-C`
Only process complete files. Files whose filename *end-time* timestamp is `not_terminated` are not processed (such a file is currently being written to by the audit system). This is useful in preventing the loss of records if `-D` is used to delete the input files after they are processed. It does not apply to files specified on the command line.

`-D` *suffix*
Delete input files after they are read if the entire run is successful. If `auditreduce` detects an error while reading a file, then that file is not deleted. If `-D` is specified, `-A`, `-C` and `-O` are also implied. *suffix* is given to the `-O` option. This helps prevent the loss of audit records by ensuring that all of the records are written, only complete files are processed, and the records are written to a file before being deleted. Note that if both `-D` and `-O` are specified in the command line, the order of specification is significant. The *suffix* associated with the latter specification is in effect.

`-M` *machine*
Allows selection of records from files with *machine* as the filename suffix. If `-M` is not specified, all files are processed regardless of suffix. `-M` can also be used to allow selection of records from files that contain combined records from many machines and have a common suffix (such as `all`).

`-N`
Select objects in *new mode*.This flag is off by default, thus retaining backward compatibility. In the existing, *old mode*, specifying the `-e`, `-f`, `-g`, `-r`, or `-u` flags would select not only actions taken with those IDs, but also certain objects owned by those IDs. When running in *new mode*, only actions are selected. In order to select objects, the `-o` option must be used.

`-O` *suffix*
Direct output stream to a file in the current `audit_root_dir` with the indicated suffix. *suffix* can alternatively contain a full pathname, in which case the last component is taken as the suffix, ahead of which the timestamps will be placed, ahead of which the remainder of the pathname will be placed. If the `-O` option is not specified, the output is sent to the standard output. When `auditreduce` places timestamps in the filename, it uses the times of the first and last records in the merge as the *start-time* and *end-time*.

`-Q`
Quiet. Suppress notification about errors with input files.

`-R` *pathname*
Specify the pathname of an alternate audit root directory *audit_root_dir* to be *pathname*. Therefore, rather than using `/var/audit` by default, *pathname*`/*` will be examined instead.

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

-S *specific_directory*
This option causes `auditreduce` to read audit trail files from a specific location (*specific_directory*). *specific_directory* is normally interpreted as the name of a subdirectory of the audit root, therefore `auditreduce` will look in *audit_root_dir*/*specific_directory* for the audit trail files. But if *specific_directory* contains any backslash characters (`/`), it is the name of a directory not necessarily contained in the audit root. In this case, *specific_directory* will be consulted. This option allows archived files to be manipulated easily, without requiring that they be physically located in a directory structure like that of `/var/audit`.

-V
Verbose. Display the name of each file as it is opened, and how many records total were written to the output stream.

Record Selection Options

The record selection options listed below are used to indicate which records are written to the output file produced by `auditreduce`.

Multiple arguments of the same type are not permitted.

-a *date-time*
Select records that occurred at or after *date-time*. The *date-time* argument is described under `Option Arguments`, below. *date-time* is in local time. The `-a` and `-b` options can be used together to form a range.

-b *date-time*
Select records that occurred before *date-time*.

-c *audit-classes*
Select records by audit class. Records with events that are mapped to the audit classes specified by *audit-classes* are selected. Audit class names are defined in `audit_class(4)`. The *audit-classes* can be a comma separated list of `audit` *flags* like those described in `audit_flags(5)`. Using the `audit` *flags,* one can select records based upon success and failure criteria.

-d *date-time*
Select records that occurred on a specific day (a 24-hour period beginning at 00:00:00 of the day specified and ending at 23:59:59). The day specified is in local time. The time portion of the argument, if supplied, is ignored. Any records with timestamps during that day are selected. If any hours, minutes, or seconds are given in *time,* they are ignored. `-d` can not be used with `-a` or `-b`.

-e *effective-user*
Select records with the specified *effective-user*.

-f *effective-group*
Select records with the specified *effective-group*.

-g *real-group*
Select records with the specified *real-group*.

`-j` *subject-ID*
Select records with the specified *subject-ID* where *subject-ID* is a process ID.

`-l` *label*
Select records with the specified label (or label range), as explained under "Option Arguments," below. This option is available only if the system is configured with Trusted Extensions.

`-m` *event*
Select records with the indicated *event*. The *event* is the literal string or the *event* number.

`-o` *object_type=objectID_value*
Select records by object type. A match occurs when the record contains the information describing the specified *object_type* and the object ID equals the value specified by *objectID_value*. The allowable object types and values are as follows:

`file=`*pathname*
Select records containing file system objects with the specified pathname, where pathname is a comma separated list of regular expressions. If a regular expression is preceded by a tilde (~), files matching the expression are excluded from the output. For example, the option `file=~/usr/openwin,/usr,/etc` would select all files in `/usr` or `/etc` except those in `/usr/openwin`. The order of the regular expressions is important because auditreduce processes them from left to right, and stops when a file is known to be either selected or excluded. Thus the option `file= /usr`, `/etc`, `~/usr/openwin` would select all files in `/usr` and all files in `/etc`. Files in `/usr/openwin` are not excluded because the regular expression `/usr` is matched first. Care should be given in surrounding the *pathname* with quotes so as to prevent the shell from expanding any tildes.

`filegroup=`*group*
Select records containing file system objects with *group* as the owning group.

`fileowner=`*user*
Select records containing file system objects with *user* as the owning user.

`fmri=`*service_instance*
Select records containing fault management resource identifier (FMRI) objects with the specified *service instance*. See `smf(5)`.

`msgqid=`*ID*
Select records containing message queue objects with the specified *ID* where *ID* is a message queue ID.

`msgqgroup=`*group*
Select records containing message queue objects with *group* as the owning or creating group.

`msgqowner=`*user*
Select records containing message queue objects with *user* as the owning or creating user.

`pid=`*ID*
Select records containing process objects with the specified *ID* where *ID* is a process ID. Process are objects when they are receivers of signals.

`procgroup=`*group*
Select records containing process objects with *group* as the real or effective group.

`procowner=`*user*
Select records containing process objects with *user* as the real or effective user.

`semid=`*ID*
Select records containing semaphore objects with the specified *ID* where *ID* is a semaphore ID.

`semgroup=`*group*
Select records containing semaphore objects with *group* as the owning or creating group.

`semowner=`*user*
Select records containing semaphore objects with *user* as the owning or creating user.

`shmid=`*ID*
Select records containing shared memory objects with the specified *ID* where *ID* is a shared memory ID.

`shmgroup=`*group*
Select records containing shared memory objects with *group* as the owning or creating group.

`shmowner=`*user*
Select records containing shared memory objects with *user* as the owning or creating user.

`sock=`*port_number*|*machine*
Select records containing socket objects with the specified *port_number* or the specified *machine* where *machine* is a machine name as defined in `hosts(4)`.

`user=`*user_name*
Select records containing the user object whose name is specified. User objects are generally specified for administrative actions on a user.

`-r` *real-user*
Select records with the specified *real-user*.

`-s` *session-id*
Select audit records with the specified *session-id*.

`-u` *audit-user*
Select records with the specified *audit-user*.

`-z` *zone-name*
Select records from the specified zone name. The zone name selection is case-sensitive.

When one or more *filename* arguments appear on the command line, only the named files are processed. Files specified in this way need not conform to the audit trail filename format. However, `-M`, `-S`, and `-R` must not be used when processing named files. If the *filename* is "–" then the input is taken from the standard input.

Option Arguments

audit-trail-file

An audit trail file as defined in `audit.log(4)`. An audit trail file not named on the command line must conform to the audit trail file name format. Audit trail files produced as output of `auditreduce` are in this format as well. The format is:

start-time . *end-time* . *suffix*

start-time is the 14 character time stamp denoting when the file was opened. *end-time* is the 14 character time stamp denoting when the file was closed. *end-time* can also be the literal string `not_terminated`, indicating the file is still be written to by the audit daemon or the file was not closed properly (a system crash or abrupt halt occurred). *suffix* is the name of the machine that generated the audit trail file (or some other meaningful suffix; for example, `all` would be a good suffix if the audit trail file contains a combined group of records from many machines).

date-time

The *date-time* argument to `-a`, `-b`, and `-d` can be of two forms: An absolute *date-time* takes the form:

yyyymmdd [*hh* [*mm* [*ss*]]]

where *yyyy* specifies a year (with 1970 as the earliest value), *mm* is the month (01-12), `dd` is the day (01-31), *hh* is the hour (00-23), *mm* is the minute (00-59), and *ss* is the second (00-59). The default is 00 for *hh*, *mm* and *ss*.

An offset can be specified as: +*n* `d|h|m| s` where *n* is a number of units, and the tags `d`, `h`, `m`, and `s` stand for days, hours, minutes and seconds, respectively. An offset is relative to the starting time. Thus, this form can only be used with the `-b` option.

event

The literal string or ordinal event number as found in `audit_event(4)`. If *event* is not found in the `audit_event` file it is considered invalid.

group

The literal string or ordinal group ID number as found in `group(4)`. If *group* is not found in the `group` file it is considered invalid. *group* can be negative.

label

The literal string representation of a MAC label or a range of two valid MAC labels. To specify a range, use `x;y` where `x` and `y` are valid MAC labels. Only those records that are fully bounded by `x` and `y` will be selected. If `x` or `y` is omitted, the default uses `ADMIN_LOW` or `ADMIN_HIGH` respectively. Notice that quotes must be used when specifying a range.

pathname
A regular expression describing a pathname.

user
The literal username or ordinal user ID number as found in passwd(4). If the username is not found in the passwd file it is considered invalid. *user* can be negative.

使用例

例 1 Using auditreduce

praudit(1M) is available to display audit records in a human-readable form.

This will display the entire audit trail in a human-readable form:

```
% auditreduce | praudit
```

If all the audit trail files are being combined into one large file, then deleting the original files could be desirable to prevent the records from appearing twice:

```
% auditreduce -V -D /var/audit/combined/all
```

This displays what user milner did on April 13, 1988. The output is displayed in a human-readable form to the standard output:

```
% auditreduce -d 19880413 -u milner | praudit
```

The above example might produce a large volume of data if milner has been busy. Perhaps looking at only login and logout times would be simpler. The -c option will select records from a specified class:

```
% auditreduce -d 19880413 -u milner -c lo | praudit
```

To see milner's login/logout activity for April 13, 14, and 15, the following is used. The results are saved to a file in the current working directory. Notice that the name of the output file will have milnerlo as the *suffix*, with the appropriate timestamp prefixes. Notice also that the long form of the name is used for the -c option:

```
% auditreduce -a 19880413 -b +3d -u milner -c login_logout -O milnerlo
```

To follow milner's movement about the file system on April 13, 14, and 15 the chdir record types could be viewed. Notice that in order to get the same time range as the above example we needed to specify the -b time as the day *after* our range. This is because 19880416 defaults to midnight of that day, and records before that fall on 0415, the end-day of the range.

```
% auditreduce -a 19880413 -b 19880416 -u milner -m AUE_CHDIR | praudit
```

In this example, the audit records are being collected in summary form (the login/logout records only). The records are being written to a summary file in a different directory than the normal audit root to prevent the selected records from existing twice in the audit root.

```
% auditreduce -d 19880330 -c lo -O /var/audit/audit_summary/logins
```

例 1 Using auditreduce (続き)

If activity for user ID 9944 has been observed, but that user is not known to the system administrator, then the command in the following example searches the entire audit trail for any records generated by that user. `auditreduce` queries the system about the current validity of ID 9944 and displays a warning message if it is not currently active:

```
% auditreduce -O /var/audit/audit_suspect/user9944 -u 9944
```

To get an audit log of only the global zone:

```
% auditreduce -z global
```

ファイル

`/var/audit/*` default location of audit trails, when stored

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

The command invocation is Stable. The binary file format is Stable. The binary file contents is Unstable.

関連項目

`praudit(1M)`, `audit.log(4)`, `audit_class(4)`, `group(4)`, `hosts(4)`, `passwd(4)`, `attributes(5)`, `audit_flags(5)`, `smf(5)`

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

診断

`auditreduce` displays error messages if there are command line errors and then exits. If there are fatal errors during the run, `auditreduce` displays an explanatory message and exits. In this case, the output file might be in an inconsistent state (no trailer or partially written record) and `auditreduce` displays a warning message before exiting. Successful invocation returns `0` and unsuccessful invocation returns `1`.

Since `auditreduce` might be processing a large number of input files, it is possible that the machine-wide limit on open files will be exceeded. If this happens, `auditreduce` displays a message to that effect, give information on how many file there are, and exit.

If `auditreduce` displays a record's timestamp in a diagnostic message, that time is in local time. However, when filenames are displayed, their timestamps are in UTC.

使用上の留意点

Conjunction, disjunction, negation, and grouping of record selection options should be allowed.

注意事項

The -z option should be used only if the audit policy zonename is set. If there is no zonename token, then no records will be selected.

名前

auditstat – display kernel audit statistics

形式

```
auditstat [-c count] [-h numlines] [-i interval] [-n]
     [-T u | d ] [-v]
```

機能説明

`auditstat` displays kernel audit statistics. The fields displayed are as follows:

- `aud` The total number of audit records processed by the userland audit.
- `ctl` This field is obsolete.
- `drop` The total number of audit records that have been dropped. Records are dropped according to the kernel audit policy. See `auditconfig(1M)`, `AUDIT_CNT` policy for details.
- `enq` The total number of audit records put on the kernel audit queue.
- `gen` The total number of audit records that have been constructed (not the number written).
- `kern` The total number of audit records produced by user processes (as a result of system calls).
- `mem` The total number of Kbytes of memory currently in use by the kernel audit module.
- `nona` The total number of non-attributable audit records that have been constructed. These are audit records that are not attributable to any particular user.
- `rblk` The total number of times that the audit queue has blocked waiting to process audit data.
- `tot` The total number of Kbytes of audit data written to the audit trail.
- `wblk` The total number of times that user processes blocked on the audit queue at the high water mark.
- `wrtn` The total number of audit records written. The difference between `enq` and `wrtn` is the number of outstanding audit records on the audit queue that have not been written.

オプション

- `-c` *count* Display the statistics a total of *count* times. If *count* is equal to zero, statistics are displayed indefinitely. A time interval must be specified.
- `-h` *numlines* Display a header for every *numlines* of statistics printed. The default is to display the header every 20 lines. If *numlines* is equal to zero, the header is never displayed.
- `-i` *interval* Display the statistics every *interval* where *interval* is the number of seconds to sleep between each collection.
- `-n` Display the number of kernel audit events currently configured.
- `-T u|d` Display a time stamp.

Specify u for a printed representation of the internal representation of time. See time(2). Specify d for standard date format. See date(1).

-v Display the version number of the kernel audit module software.

終了ステータス

auditstat returns 0 upon success and 1 upon failure.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

auditconfig(1M), praudit(1M), attributes(5)

名前

audit_warn – audit daemon warning script

形式

`/etc/security/audit_warn [option [arguments]]`

機能説明

The `audit_warn` utility processes warning or error messages from the audit daemon. When a problem is encountered, the audit daemon, `auditd(1M)` calls `audit_warn` with the appropriate arguments. The *option* argument specifies the error type.

The system administrator can specify a list of mail recipients to be notified when an audit_warn situation arises by defining a mail alias called `audit_warn` in `aliases(4)`. The users that make up the `audit_warn` alias are typically the `audit` and `root` users.

オプション

The following options are supported:

`allhard` *count*
: Indicates that the hard limit for all filesystems has been exceeded *count* times. The default action for this option is to send mail to the `audit_warn` alias only if the *count* is `1`, and to write a message to the machine console every time. It is recommended that mail *not* be sent every time as this could result in a the saturation of the file system that contains the mail spool directory.

`allsoft`
: Indicates that the soft limit for all filesystems has been exceeded. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`auditoff`
: Indicates that someone other than the audit daemon changed the system audit state to something other than `AUC_AUDITING`. The audit daemon will have exited in this case. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`ebusy`
: Indicates that the audit daemon is already running. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`hard` *filename*
: Indicates that the hard limit for the file has been exceeded. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`hostname`
: Indicates that the audit service could not find an IP address to associate with the local hostname. It has fallen back to using the “loopback” address. The audit trail might not

translate properly. See `/var/audit/debug` for more information. The audit service can be refreshed (`audit -s`) to retry to find an IP address.

`nostart`

Indicates that auditing could not be started. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console. Some administrators may prefer to modify `audit_warn` to reboot the system when this error occurs.

`plugin` *name error count text*

Indicates that an error occurred during execution of the `auditd` plugin *name*. The default action for this option is to send mail to the `audit_warn` alias only if *count* is 1, and to write a message to the machine console every time. (Separate counts are kept for each error type.) It is recommended that mail not be sent every time as this could result in the saturation of the file system that contains the mail spool directory. The *text* field provides the detailed error message passed from the plug-in. The *error* field is one of the following strings:

`load_error`

Unable to load the plugin *name*.

`sys_error`

The plugin *name* is not executing due to a system error such as a lack of resources.

`config_error`

No plug-ins loaded (including the binary file plug-in, `audit_binfile(5)`) due to configuration errors (see the `-setplugin` option of the `auditconfig(1M)` command). The name string is `--` , to indicate that no plug-in name applies.

`retry`

The plugin *name* reports it has encountered a temporary failure. For example, the `audit_binfree.so` plugin uses `retry` to indicate that all directories are full.

`no_memory`

The plugin *name* reports a failure due to lack of memory.

`invalid`

The plugin *name* reports it received an invalid input.

`failure`

The plugin *name* has reported an error as described in *text*.

`postsigterm`	Indicates that an error occurred during the orderly shutdown of the audit daemon. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.
`soft` *filename*	Indicates that the soft limit for *filename* has been exceeded. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.
`tmpfile`	Indicates that the temporary audit file already exists indicating a fatal error. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

The interface stability is Evolving. The file content is Unstable.

関連項目

`audit(1M)`, `auditconfig(1M)`, `auditd(1M)`, `aliases(4)`, `audit.log(4)`, `attributes(5)`

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

注意事項

This functionality is available only if the Solaris Auditing feature has been enabled.

If the audit policy `perzone` is set, the `/etc/security/audit_warn` script for the local zone is used for notifications from the local zone's instance of `auditd`. If the `perzone` policy is not set, all `auditd` errors are generated by the global zone's copy of `/etc/security/audit_warn`.

名前

automount – 自動マウントポイントのインストール

形式

/usr/sbin/automount [-t *duration*] [-v]

機能説明

automount ユーティリティーは、autofs マウントポイントをインストールし、automount マップと各マウントポイントを対応付けます。ローカルまたは分散された automount マップのどちらかに重要なエントリがあり、automountd(1M) デーモンがまだ起動していない場合、automount ユーティリティーは automountd1M デーモンを起動します。autofs ファイルシステムは、自身のファイルシステム内のディレクトリへのアクセスを監視し、automountd(1M) デーモンに通知します。このデーモンはマップを使用してファイルシステムを特定し、autofs ファイルシステム内の参照ポイントにそのファイルシステムをマウントします。マップは、/etc/auto_master マップのエントリを使用するか、または直接マップを使用することによって、autofs マウントに割り当てることができます。

一定時間内 (デフォルトでは 10 分間) にファイルシステムに対してアクセスがなければ、automountd デーモンはファイルシステムのマウントを解除します。

ファイル /etc/auto_master により、すべての autofs マウントポイントの位置が決定されます。デフォルトでは、このファイルには 3 つのエントリが含まれます。

```
# Master map for automounter
#
+auto_master
/net          -hosts    -nosuid
/home         auto_home
```

+auto_master エントリは、外部 NIS マスターマップへの参照です。マスターマップが存在する場合、+auto_master エントリの位置へ読み取られます。マスターファイルの残りのエントリは、autofs マウントが行われるディレクトリを指定し、さらにそのディレクトリに対応付けるオートマウントマップを指定します。各エントリの 3 番目のオプションフィールドには、任意のマウントオプションを指定できます。これらのオプションは、マウントオプションが明示的に指定されていない、マップ内のすべてのエントリに適用されます。automount コマンドは通常、引数を指定しないで実行します。automount コマンドは、/etc/auto_master 内のエントリと /etc/mnttab 内の autofs マウントのその時点のリストを比較し、autofs マウントを追加、削除、または変更することによって、/etc/auto_master の情報を /etc/mnttab ファイルに反映させます。automount コマンドはブート時に、マスターマップからすべての autofs マウントをインストールします。その後、マスターマップまたは直接マップの新規エントリを autofs マウントにインストールするか、削除されたエントリに関してのマウントを解除します。

SMF による管理

automount サービスはサービス管理機能 smf(5) により次のサービス識別子の下で管理されます。

```
svc:/system/filesystem/autofs:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

automount の起動時 autofs SMF パラメータは、sharectl(1M) を使用して操作できます。これらのパラメータを表示および更新する方法については、sharectl(1M) を参照してください。サポートされるパラメータは次のとおりです。

timeout=*num*
: ファイルシステムが使用されていないときに、マウントしたままにしておく時間の間隔を秒数で指定します。デフォルト値は 600 (10 分) です。automount の -t オプションと同等の機能です。

automount_verbose=TRUE | FALSE
: 冗長モード。autofs のマウントおよびマウント解除のような重要ではないイベントを通知します。デフォルト値は FALSE です。automount の -v オプションと同等の機能です。

Oracle Solaris Trusted Extensions でのオートマウント

システムが Oracle Solaris Trusted Extensions で構成されている場合、ユーザーには認可上限内の各ラベルにホームディレクトリがあります。このため、対応するラベル付きの各ゾーンでホームディレクトリが使用可能である必要があります。auto_home マップを使用する代わりに、特殊マップが自動的に生成され、ゾーンの名前が接尾辞として付けられます。デフォルトでは、このマップには単一のエントリが含まれます。

```
-fstype=lofs   :/export/home/&
```

ホームディレクトリが参照された際に、その名前がゾーンの auto_home_*zonename* マップ内のほかのどのキーとも一致しなかった場合、それはこのループバックマウント指定に一致します。このループバック一致が発生し、かつその名前が、ゾーン内にホームディレクトリを持たない有効なユーザーに対応していた場合、ユーザーに代わってディレクトリが自動的に作成されます。

NFS を使用すると、ゾーン内のホームディレクトリを、上位レベルのゾーンと読み取り専用モードで共有することもできます。この場合、上位レベルのゾーンは、インポートされる下位レベルの各ゾーンのオートマップエントリを備えている必要があります。内部ゾーンで解釈される、公開ゾーンの一般的なマップエントリは、auto_home_public と呼ばれ、次のような内容を含んでいます。

```
+auto_home_public

public-zone-IP-address:/export/home/&
```

このオートマップエントリは、次のように /etc/auto_master に含められます。

```
/zone/public/home      auto_home_public      -nobrowse
```

上位レベルのゾーンのユーザーは、updatehome(1) ユーティリティーを使用すると、最下位のラベル付きゾーンを同期元として使用し、特定の起動ファイルを同期できます。

オプション

次のオプションがサポートされています。

-t *duration*　ファイルシステムが使用されていないときに、マウントしたままにしておく時間の間隔を秒数で指定します。デフォルトは 10 分です。

-v　冗長モード。autofs のマウント、マウント解除、またはその他の関連情報を通知します。

使用法

マップエントリの形式

単純マップエントリ (マッピング) の形式は、次のとおりです。

key [-*mount-options*] *location* . . .

key は、直接マップで使用する場合はマウントするディレクトリのフルパス名であり、間接マップの場合はサブディレクトリからの相対名です。*mount-options* は、コンマで区切った mount オプションのリストです。*location* には、ファイルシステムがマウントされるディレクトリを指定します。単純 NFS マウントの場合は、mount_nfs(1M) で指定されているのと同じオプションが使用できます。*location* の形式は次のとおりです。

host: *pathname*

host は、マウントするファイルシステムが置かれているホスト名です。*pathname* は、マウントするディレクトリの絶対パス名です。

ほかのファイルシステム用のオプションについては、mount_nfs(1M) など、ほかの mount_* のマニュアルページを参照してください。

複製されたファイルシステム

複製された NFS ファイルシステムには複数の *location* フィールドを指定できます。その場合、 automount とカーネルのそれぞれがその情報を使用して、可用性の向上を計ります。読み取り専用フラグがマップエントリに設定されている場合、automountd はカーネルが使用する、いくつかの条件によってソートされている位置リストを使用してマウントします。マウント時に利用できる位置だけがマウントされ、カーネルで利用可能になります。サーバーが応答しなかった場合、カーネルは代替サーバーに切り替えます。automount のソート順序は、次のサーバーの選択方法を決定するのに使用されます。読み取り専用フラグが設定されていない場合、automount は同じソート順序によって選択された、最善の位置をマウントします。新しいサーバーが選択されるのは、マウント解除が可能だった場合に限られ、再マウントが実行されます。同じローカルサブネット上のサーバーに最も高い優先順位が与えられ、その次にローカルネット上のサーバーに高い優先順位が与えられます。等距離のサーバー間では、重み付け係数 (後述) が使用されていないかぎり、応答時間によって順序が決まります。

NFS Version 2 プロトコルと NFS Version 3 プロトコルの両方を使用するサーバーの位置がリストに含まれている場合、automount はリストに指定されたサーバー位置の一部だけを選択し、すべてのエントリが同じプロトコルになるようにします。 NFS

Version 3 プロトコルを使用するサーバーが選択されるのは、ローカルサブネット上にある、NFS Version 2 プロトコルを使用するサーバーが無視されない場合に限られます。詳細については、『Oracle Solaris の管理: IP サービス』を参照してください。

リストの各 *location* が同じ *pathname* を共有している場合、コンマで区切ったホスト名に対して、1 つの *location* を使用できます。

```
hostname,hostname . . . : pathname
```

括弧で囲んだ整数を重み付け係数としてサーバー名のうしろに指定することにより、サーバー要求に重み付けを設定できます。重み付けが設定されていないサーバーは、ゼロの値 (選択される可能性が最も高い) が与えられているものとみなされます。値が大きくなるにしたがって、選択される可能性が小さくなります。次に例を示します。

```
man -ro alpha,bravo,charlie(1),delta(4) : /usr/man
```

ホスト alpha および bravo に最も高い優先順位が与えられており、ホスト delta には最も低い優先順位が与えられています。

選択プロセスではサーバーの近接度が優先されます。前述の例で、サーバー delta はクライアントと同じネットワークセグメントにあり、ほかのサーバーは異なるネットワークセグメントにある場合、重み付けの値は無視され、delta が選択されます。重み付けが有効なのは、同じネットワーク近接度を持つサーバー間で選択が行われる場合だけです。自動マウンターは重み付けに関わらず、同じネットワークセグメント上の他のサーバー経由でローカルホストをつねに選択します。

サーバーごとにエクスポートポイントが異なる場合にも、重み付けを適用できます。例:

```
man -ro alpha:/usr/man  bravo,charlie(1):/usr/share/man
     delta(3):/export/man
```

マッピングを複数の入力行にわたって記述する場合は、バックスラッシュ (\) で復帰改行をエスケープします。コメントは番号記号 (#) で始まり、その後の復帰改行で終了します。

マップキーの置換

アンパサンド (&) 記号は、使用されているエントリの key フィールドの値に展開されます。次のコマンドを見てください。

```
jane sparcserver : /home/&
```

この場合、& は jane に展開されます。

ワイルドカードキー

アスタリスク (*) 記号は、key フィールドとして与えられた場合、どのようなキーにも当てはまるエントリとして認識されます。このようなエントリは、それまで一致しなかったあらゆるキーと一致します。たとえば、/config の間接マップに次のエントリが指定されている場合、

```
*         & : /export/config/&
```

位置を次のように指定できるリモートファイルシステムは、/config にすべて自動マウントされます。

```
hostname : /export/config/hostname
```

ワイルドカードキーは -browse オプションと組み合わせて使用できないことに注意してください。

変数置換

automount マップ内でクライアント固有の変数を使用できます。たとえば、マップに $HOST が指定されている場合、automount は $HOST をクライアントのホスト名に対応する現在値に展開します。次の変数を使用できます。

ARCH	arch の出力	アーキテクチャ名。たとえば、 sun4u マシンでは sun4。
CPU	uname -p の出力	プロセッサタイプ たとえば、sparc
HOST	uname -n の出力	ホスト名 たとえば、myhost。
KARCH	arch -k または uname -m の出力	カーネルアーキテクチャー名またはマシンのハードウェア名。たとえば、sun4u。
OSNAME	uname -s の出力	OS 名 たとえば、SunOS
OSREL	uname -r の出力	OS リリース名 たとえば、5.3
OSVERS	uname -v の出力	OS バージョン たとえば、beta1.0
NATISA	isainfo -n の出力	システムのネイティブ命令セットアーキテクチャー。 たとえば、sparcv9
PLATFORM	uname -i の出力	プラットフォーム名。たとえば、SUNW,Sun-Fire-V240。

参照が接辞付き文字から保護される必要がある場合は、変数名を中括弧 ({ }) で囲みます。

多重マウント

多重マウントエントリの形式は、次のとおりです。

```
key [-mount-options] [ [mountpoint] [-mount-options] location. . . ] . . .
```

最初の /[*mountpoint*] は、最初のマウントでは省略可能ですが、以降のすべてのマウントでは必須です。省略可能な *mountpoint* は、key で指定されたディレクトリからの相対パス名とみなされます。最初の *mountpoint* を省略した場合、/ (ルート) の *mountpoint* が暗黙に使用されます。

/src の直接マップに次のエントリがあるとします。

```
beta     -ro\
  /           svr1,svr2:/export/src/beta  \
  /1.0        svr1,svr2:/export/src/beta/1.0  \
  /1.0/man    svr1,svr2:/export/src/beta/1.0/man
```

すべてのオフセットは、beta の下のサーバー上になければなりません。automount は svr1 または svr2 のどちらかから、つまり最も近くにあり先に応答した方のホストから、/src/beta、/src/beta/1.0、および /src/beta/1.0/man を必要に応じて自動的にマウントします。

その他のファイルシステムタイプ

オートマウントは、デフォルトのファイルシステムタイプとして NFS マウントを想定しています。他のファイルシステムタイプを指定する場合は、fstype マウントオプションを使用します。fstype オプションと組み合わせることにより、そのファイルシステムタイプ固有の他のマウントオプションを使用できます。位置フィールドにはファイルシステムタイプ固有の情報を指定する必要があります。たとえば、CD ファイルシステムをマウントする場合のように、位置フィールドがスラッシュで始まる場合は、その前にコロンが必要です。

```
cdrom -fstype=hsfs,ro   : /dev/sr0
```

autofs マウントを実行するには、次のように指定します。

```
src -fstype=autofs    auto_src
```

注: この手順は、Volume Manager を使用しない場合に限って使用してください。

オプションの継承については、「注意事項」を参照してください。

間接マップ

間接マップを使用すると、コマンド行で指定した directory からマウントするサブディレクトリの、マッピングを指定できます。間接マップの各 key は、必要に応じてマウントする 1 つ以上のファイルシステムを参照する単純名です。

直接マップ

直接マップのエントリは、autofs マウントポイントと直接対応づけられます。各 *key* は autofs マウントポイントのフルパス名です。直接マップ全体が 1 つのディレクトリに対応付けられることはありません。

直接マップは、/- キーによって間接マップと区別されます。例:

```
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/-              auto_direct
```

インクルードマップ

次の形式のエントリを使用すると、マップ内に別のマップの内容を含めることができます。

+*mapname*

mapname がスラッシュで始まる場合、mapname はローカルファイルのパス名とみなされます。それ以外の場合、マップの位置は次に示すように、/etc/nsswitch.conf 内のオートマウンタ用のエントリに基づくネームサービススイッチのポリシーによって決まります。

```
automount: files nis
```

ネームサービスが files の場合、/etc 内のローカルファイルの名前が想定されます。検索対象のキーが取り込まれたマップにない場合、次のエントリで検索が続けられます。

特殊マップ

使用可能な特殊マップには 3 つの種類があります。-hosts、-fedfs、および -null です。hosts マップは /net ディレクトリと組み合わせて使用します。この場合、マップのキーは NFS サーバーのホスト名とみなされます。automountd デーモンは、エクスポートされたファイルシステムのサーバーリストに基づいて、マップエントリを動的に作成します。/net/hermes のディレクトリ参照では、hermes のルートからの相対位置にある、対応するディレクトリが参照されます。

-fedfs マップは /nfs4 ディレクトリと組み合わせて使用します。この場合、マップのキーはドメインのルートファイルシステムが必要な DNS ドメインとみなされます。automountd デーモンは、ドメインのルートサーバーを次のようなクエリーで検索します。

```
% nslookup -q=srv _nfs4._domainroot._tcp.domain
```

その後、*server-list*:/.domainroot-*domain* を /nfs4/*domain* にマウントします。

これは、次の文書にドキュメント化されている保留中の IETF 標準をサポートします。

```
http://datatracker.ietf.org/doc/\
draft-ietf-nfsv4-federated-fs-dns-srv-namespace/
```

-null マップは、指定されたディレクトリに対してすでに設定されているマップを取り消します。これは /etc/auto_master において、+auto_master によって与えられるエントリから継承されるエントリを取り消す場合に便利です。ただし、取り込まれたマップのエントリの前に -null エントリを挿入しないと、効果はありません。

実行可能マップ

ファイルパーミションの実行ビットが設定されているローカルマップは、オートマウンタによって実行され、引数として検索されるキーを与えられます。実行可能マップは、標準出力にオートマウンタのマップエントリの内容を返すようにします。エントリが特定できなかった場合、出力はありません。直接マップを実行可能にすることはできません。

構成と auto_master マップ

引数を指定しないで automount を開始した場合、automount はマスターマップで、autofs マウントポイントのリストおよび対応するマップを調べます。automount はまだマウントされていない autofs マウントがあればそれらをマウントし、マスターマップまたは直接マップから削除されている autofs マウントがあればそれらをマウント解除します。

マスターマップは auto_master とみなされ、その位置はネームサービススイッチのポリシーによって決まります。通常、ローカルファイル /etc/auto_master がマスターマップとして最初に検索されます。

ブラウズ機能

automount デーモンは、間接マップのブラウズ機能をサポートします。これにより、マウントされているかどうかに関係なく、マウント可能なマウントポイントをすべて表示できます。任意の間接 autofs マップに -nobrowse オプションを追加すると、ブラウズ機能を無効にできます。例:

```
/net     -hosts      -nosuid,nobrowse
/home    auto_home
```

この場合、/net で表示されるのはマウントされている *hostnames* だけですが、/homeではマウント可能なすべてのマウントポイントが表示されます。autofs ファイルシステムのブラウズ機能は、-browse オプションによって有効になります。これはすべての間接マップにおけるデフォルトの設定です。

-browse オプションは、ワイルドカードキーと組み合わせては使用できません。

マウント制限マップ

あるマップに指定したオプションは、そのマップ内のすべてのエントリに対するデフォルトのオプションとして使用されます。そして、このようなオプションは、マップ内のエントリが独自のマウントオプションを指定するときに初めて無視されます。

しかし、マウントマップとそのサブマウント全体に、nosuid、nodevices、nosetuid、または noexec を強制的に適用することが望ましい場合もあります。これを行うには、拡張マウントオプション -restrict を使用します。

```
/home     auto_home       -restrict,nosuid,hard
```

-restrict オプションを使用すると、制限的なオプション (つまり、nosuid、nodevices、nosetuid、および noexec) はすべて、-restrict オプション自身とともに、強制的に継承されます。前述の例では、nosuid と restrict オプションは継承されますが、hard オプションは継承されません。また、restrict オプションは「実行可能なマップ」の実行を防ぎます。さらに、restrict オプション

は、あるプログラムが自分のゾーン内で利用できるすべての特権を持たずに、一部の特権だけで確立した自動マウントに対しても強制されます。

終了ステータス

次の終了値が返されます。

0　正常終了。

1　エラーが発生しました。

ファイル

/etc/auto_master	マスターオートマウントマップ
/etc/auto_home	オートマウントされるホームディレクトリ用のマップ
/etc/nsswitch.conf	ネームサービススイッチの構成ファイル。nsswitch.conf(4) を参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

isainfo(1), ls(1), svcs(1), uname(1), updatehome(1), automountd(1M), mount(1M), mount_nfs(1M), sharectl(1M), svcadm(1M), attributes(5), nfssec(5), smf(5)

『Oracle Solaris のシステム管理 (ネットワークサービス)』

注意事項

autofs マウントポイント間に階層関係が存在してはいけません。automount では、autofs マウント内に別の autofs マウントポイントを作成できません。

直接マップエントリごとに新しい autofs マウントが作成されるので、直接マップはできるだけ短くしてください。

直接マップと間接マップのエントリはどちらも、いつでも変更できます。新しい情報は、automountd が次回そのマップエントリを使用してマウントを実行するときに使用されます。

マスターマップまたは直接マップに追加された新しいエントリは、automount コマンドを実行してそれらの新しいエントリを新しい autofs マウントポイントとしてインストールするまで使用できません。間接マップに追加された新しいエントリは、ただちに使用できます。

Solari 2.6 リリースでは、間接マップに対応づけられた autofs ディレクトリの表示 (ls(1) を参照) には、潜在的にマウント可能なエントリがすべて含まれます。潜在的にマウント可能なエントリに対応づけられる属性は一時的なものです。実際のファイルシステム属性が表示されるのは、ファイルシステムのマウント後に限られます。

マスターマップの 3 番目のオプションフィールドを指定すると、デフォルトのマウントオプションをマップ全体に割り当てることができます。これらのオプションが適用されるのは、マウントオプションが指定されていないマップエントリだけです。マップエントリにオプションが指定されていると、デフォルトのオプションは無効になります。現時点では、オプションの連結は行われません。連結機能は、将来のリリースに備えて計画中です。

オートマウンタのデフォルトの NFS マウント操作の再試行回数は 0 です。つまり、マウント試行は 1 回だけで、再試行されません。これは、`mount_nfs(1M)` ユーティリティーのデフォルト (10000 回) と大きく異なる点なので、注意してください。

ネットワーク情報サービス (NIS) は従来、Sun イエローページ (YP) と呼ばれていました。これらの機能は同等です。

名前

automountd – autofs mount/unmount daemon

形式

automountd [-Tvn] [-D *name=value*]

機能説明

automountd is an RPC server that answers file system mount and unmount requests from the autofs file system. It uses local files or name service maps to locate file systems to be mounted. These maps are described with the automount(1M) command.

If automount finds any non-trivial entries in either the local or distributed automount maps and if the daemon is not running already, the automountd daemon is automatically invoked by automount(1M). automountd enables the svc:/network/nfs/nlockmgr service (lockd(1M)), and the svc:/network/nfs/status service (statd(1M)), if NFS mounts need to be done.

SMF Management

The automountd service is managed by the service management facility, smf(5), under the service identifier:

svc:/system/filesystem/autofs

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using svcs(1). If it is disabled, it is enabled by automount(1M) unless the application/auto_enable property is set to false.

Values for parameters for automountd are stored in SMF and manipulated using sharectl(1M). The following are the supported parameters.

automountd_verbose=TRUE | FALSE
: Verbose mode. Causes status messages to be logged to:

  /var/svc/log/system-filesystem-autofs:default.log

  See smf(5). The default value is FALSE. Equivalent to the -v option.

nobrowse=ON | OFF
: Turn on or off browsing for all autofs mount points. The default value is OFF. Equivalent to the -n option.

trace=*num*
: Expands each RPC call and logs it to:

  /var/svc/log/system-filesystem-autofs:default.log

  See smf(5). The default value, 0, turns off such tracing. Starting with 1, with each higher value, the verbosity of trace output increases. This property is equivalent to the -T option.

environment=*name=value*
: Environment variables. You can specify multiple *name=value* pairs with a comma separator. If there is more than one value for a name, use the double forward slash to indicate that. See EXAMPLES. There are no environment variable settings supplied. This property is equivalent to the -D option.

All the above parameters can be changed using the sharectl(1M) command.

オプション

The following options are supported:

`-D` *name=value*	Assign *value* to the indicated `automount` map substitution variable. These assignments cannot be used to substitute variables in the master map `auto_master`.
`-n`	Turn off browsing for all `autofs` mount points. This option overrides the `-browse` `autofs` map option on the local host.
`-T`	Trace. Expand each RPC call and display it on the standard output.
`-v`	Verbose. Log status messages to the console.

使用法

See `largefile(5)` for the description of the behavior of `automountd` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

使用例

例 1 Setting Verbose Mode

The following command turns on verbose mode, causing status messages to be logged to the location specified in "SMF Management," above. See that section for a description of the properties specified in this and following examples.

```
# sharectl set -p automountd_verbose=true autofs
```

例 2 Turning on Browsing for Mount Points

The following command turns on browsing for all `autofs` mount points.

```
# sharectl set -p nobrowse=off autofs
```

例 3 Specifying and Displaying Environment Variables

The following commands set and display the value for an environment variable named `DAY`.

```
# sharectl set -p environment=DAY=TUES autofs
% sharectl get -p environment autofs
environment=DAY=TUES
```

The following command sets multiple parameters for an environment variable. The subsequent command displays the result.

```
# sharectl set -p environment=DAY=TUES,TIME=NOON autofs
% sharectl get -p environment autofs
environment=DAY=TUES,TIME=NOON
```

The following command set multiple values for name that is assigned to an environment variable. The subsequent command displays the result.

```
# sharectl set -p environment=DAY=MON\\,TUE,TIME=NOON autofs
% sharectl get -p environment autofs
environment=DAY=MON\,TUE,TIME=NOON
```

ファイル

`/etc/auto_master` Master map for automounter.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`svcs(1)`, `automount(1M)`, `sharectl(1M)`, `svcadm(1M)`, `attributes(5)`, `largefile(5)`, `smf(5)`

名前

autopush – configures lists of automatically pushed STREAMS modules

形式

`autopush -f` *filename*

`autopush -g -M` *major* `-m` *minor*

`autopush -r -M` *major* `-m` *minor*

機能説明

The `autopush` command configures the list of modules to be automatically pushed onto the stream when a device is opened. It can also be used to remove a previous setting or get information on a setting.

オプション

The following options are supported:

`-f` *filename*

Sets up the `autopush` configuration for each driver according to the information stored in *filename*. An `autopush` file consists of lines of four or more fields, separated by spaces as shown below:

major minor last-minor module1 module2 . . . module8

The first field is a string that specifies the *major* device name, as listed in the `/kernel/drv` directory. The next two fields are integers that specify the *minor* device number and *last-minor* device number. The fields following represent the names of modules. If *minor* is `–1`, then all minor devices of a major driver specified by *major* are configured, and the value for *last-minor* is ignored. If *last-minor* is `0`, then only a single minor device is configured. To configure a range of minor devices for a particular major, *minor* must be less than *last-minor*.

The remaining fields list the names of modules to be automatically pushed onto the stream when opened, along with the position of an optional anchor. The maximum number of modules that can be pushed is eight. The modules are pushed in the order they are specified. The optional special character sequence `[anchor]` indicates that a STREAMS anchor should be placed on the stream at the module previously specified in the list; it is an error to specify more than one anchor or to have an anchor first in the list.

A nonzero exit status indicates that one or more of the lines in the specified file failed to complete successfully.

`-g`

Gets the current configuration setting of a particular *major* and *minor* device number specified with the `-M` and `-m` options respectively and displays the autopush modules associated with it. It will also return the starting minor device number if the request corresponds to a setting of a range (as described with the `-f` option).

`-m` *minor*

Specifies the minor device number.

`-M` *major*

Specifies the major device number.

-r
Removes the previous configuration setting of the particular *major* and *minor* device number specified with the -M and -m options respectively. If the values of *major* and *minor* correspond to a previously established setting of a range of minor devices, where *minor* matches the first minor device number in the range, the configuration would be removed for the entire range.

終了ステータス

The following exit values are returned:

0
Successful completion.

non-zero
An error occurred.

使用例

例 1 Using the autopush command.

The following example gets the current configuration settings for the *major* and *minor* device numbers as indicated and displays the autopush modules associated with them for the character-special device /dev/term/a:

```
example# autopush -g -M 29 -m 0
Major     Minor     Lastminor     Modules
  29        0         1           ldterm  ttcompat
```

ファイル

/etc/iu.ap

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

dladm(1M), ttymon(1M), attributes(5), ldterm(7M), sad(7D), streamio(7I), ttcompat(7M)

『STREAMS Programming Guide』

注意事項

The use of the autopush command is obsolete for networking data-links. The preferred method of configuring a list of automatically pushed STREAMS modules on a given networking data-link interface is the dladm(1M) autopush link property.

Because network devices are self-cloning, the autopush command is inadequate for networking devices. The granularity of the autopush command's configuration is driver-wide, and not per-interface as one might expect. Another reason not to use autopush is that administrators are more familiar with the names of their network interfaces than with device major and minor numbers. The dladm(1M) command allows the configuration using data-link interface names.

名前

bart – 基本監査報告機能

形式

```
/usr/bin/bart create [ -n] [-R root_directory]
     [-r rules_file | -]

/usr/bin/bart create [-n] [-R root_directory] -I
     [file_name]...

/usr/bin/bart compare [-i attribute ] [-p]
     [-r rules_file | -] control-manifest test-manifest
```

機能説明

bart(1M) は、システムのソフトウェアの内容をファイルレベルでチェックするツールです。

bart_rules という規則ファイルを使って、追跡するファイルやフラグを付ける相違の種類を指定することもできます。bart_rules(4) を参照してください。

bart ユーティリティーは、次の 2 つの基本機能を実行します。

bart create

これは目録生成ツールで、システムのファイルレベルのスナップショットを取得します。出力は、「マニフェスト」と呼ばれるファイル属性のカタログです。bart_manifest(4) を参照してください。

カタログ化するファイルのリストを指定するには、3 つの方法があります。1 つ目はオプションなしの bart create を使用する方法、2 つ目はコマンド行にファイル名を指定する方法、3 つ目は監視対象のファイルを指定する指示語を含む規則ファイルを作成する方法です。bart_rules(4) を参照してください。

デフォルトでは、目録生成ツールはルート (/) ファイルシステムに含まれるすべてのファイルのすべての属性をカタログ化します。ルートファイルシステムにマウントされたファイルシステムは、ルートファイルシステムと同じ種類である場合にのみカタログ化されます。

たとえば、/、/usr、および /opt は別個の UFS ファイルシステムです。/usr と /opt は / にマウントされています。したがって、これら 3 つのファイルシステムはすべてカタログ化されます。しかし、/tmp は / にマウントされていますが、TMPFS ファイルシステムであるためカタログ化されません。マウントされた CD-ROM は、HSFS ファイルシステムであるためカタログ化されません。

bart compare

これはレポートツールで、2 つの目録を比較します。出力は、ファイルごとに属性の相違を示したリストです。これらの相違は、制御目録とテスト目録という 2 つの目録の相違です。

「相違」とは、両方の目録でカタログ化されている特定のファイルの属性に対する変更です。目録に追加されたファイルや目録から削除されたファイルは、相違として報告されます。

レポーティングメカニズムとして、詳細出力と、プログラム的な出力の 2 種類を利用できます。詳細出力は、ローカライズされ、複数の行にわたって表示されます。プログラム的な出力は、ほかのプログラムで解析しやすい形式です。「出力」を参照してください。

デフォルトでは、レポートツールは詳細出力を生成し、変更されたディレクトリのタイムスタンプ (`dirmtime` 属性) を除くすべての相違を報告します。

一貫性のある正確な比較結果を得るためには、同じ規則ファイルを使って制御目録 (*control-manifest*) とテスト目録 (*test-manifest*) を作成する必要があります。

目録を作成したり、2 つの目録を比較したりするときに、指定したファイルまたはサブツリーを無視するには、規則ファイルを使用します。ユーザーは、異なる規則ファイルを指定して `bart compare` を再実行することにより、異なる観点で目録を比較できます。

オプション

次のオプションがサポートされています。

`-i` *attribute* ...
: グローバルに無視するファイル属性を指定します。属性をコンマ区切りリストとして指定します。

  このオプションを使用すると、規則ファイルのグローバルな `IGNORE` キーワードにファイル属性を指定した場合と同じ動作になります。`bart_rules(4)` を参照してください。

`-I` [*file_name*...]
: ファイルの入力リストを指定します。このファイルリストは、コマンド行に指定することも、標準入力から読み取ることもできます。

`-n`
: ファイルリストに挙げられたすべての通常ファイルのコンテンツ署名を計算しないようにします。

`-p`
: プログラムでの解析に適した「プログラム的なモード」で目録比較出力を表示します。出力はローカライズされません。

`-r` *rules_file*
: *rules_file* を使って、カタログ化するファイルおよびディレクトリを指定し、フラグを付けるファイル属性の相違を定義します。*rules_file* が - である場合は、標準入力から規則が読み取られます。構文の定義については、`bart_rules(4)` を参照してください。

-R *root_directory*　目録のルートディレクトリを指定します。規則に指定するパスも、目録で報告されるパスも、すべて *root_directory* からの相対パスです。

注 – いかなる非大域ゾーンのルートファイルシステムも -R で参照しないでください。この操作によって、大域ゾーンのファイルシステムを損傷したり、大域ゾーンのセキュリティーを損ねたり、非大域ゾーンのファイルシステムを損傷する可能性があります。zones(5) を参照してください。

-a [*hash*]　ファイル内容のハッシュに使用する暗号化ダイジェストアルゴリズムを指定します。現在サポートされている値は、md5、sha1、sha256、sha512 です。-a を指定しない場合は、sha256 が使用され、どのハッシュアルゴリズムが使用されるかを示す Version 1.1 マニフェストが作成されます。md5 を指定した場合は、Version 1.0 マニフェストが作成されます。

オペランド

bart では、オペランドを引用符で囲むことができます。これは、指定するサブツリーやサブツリー修飾子にスペースが含まれる場合に特に重要です。

次のオペランドがサポートされています。

control-manifest　制御システム上で bart create によって作成された目録を指定します。

test-manifest　テストシステム上で bart create によって作成された目録を指定します。

出力

bart create および bart compare コマンドは、出力を標準出力に書き出し、エラーを標準エラーに書き出します。

bart create コマンドは、システム目録を生成します。bart_manifest(4) を参照してください。

bart compare コマンドは、2 つのシステム目録を比較するときに、ファイル間の相違を示すリストを生成します。デフォルトでは、比較出力はローカライズされます。ただし、-p オプションを指定すると、プログラムで操作するのに適した形式で出力が生成されます。

デフォルトの形式

filename
attribute control:*xxxx* test:*yyyy*

filename　*control-manifest* と *test-manifest* で異なるファイルの名前。スペース文字や改行文字が含まれているファイル名については、bart_manifest(4) を参照してください。

attribute 比較された目録間で相違があるファイル属性の名前。*xxxx* は *control-manifest* の属性値であり、*yyyy* は *test-manifest* の属性値です。同じファイルの複数の属性に相違が見つかった場合、別々の行にそれぞれの相違が示されます。

サポートされている属性は次のとおりです。

acl ファイルの ACL 属性。ACL 属性を持つファイルの場合、このフィールドには `acltotext()` の出力が入ります。

all すべての属性。

contents ファイルのチェックサム値。この属性が指定されるのは通常ファイルのみです。コンテキストのチェックを無効にした場合と、チェックサムが計算できない場合は、このフィールドの値は - になります。

dest シンボリックリンクのリンク先。

devnode デバイスノードの値。この属性が使用されるのは、キャラクタデバイスファイルとブロックデバイスファイルのみです。

dirmtime ディレクトリに変化が起きた時点。1970 年 1 月 1 日の 00:00:00 UTC から数えた秒数で示されます。

gid このエントリの所有者の、数値で示したグループ ID。

lnmtime リンクが作成された時点。

モード ファイルのアクセス権を示す 8 進の数値。

mtime ファイルに変化が起きた時点。1970 年 1 月 1 日の 00:00:00 UTC から数えた秒数で示されます。

size ファイルサイズ (バイト)。

type ファイルの種類。

uid このエントリの所有者の、数値で示したユーザー ID。

次のデフォルト出力は、`/etc/passwd` ファイルの属性の相違を示しています。この出力から、`size`、`mtime`、および `contents` 属性に変化があったことがわかります。

```
/etc/passwd:
  size  control:74  test:81
  mtime  control:3c165879  test:3c165979
  contents  control:daca28ae0de97afd7a6b91fde8d57afa
test:84b2b32c4165887355317207b48a6ec7
```

プログラム的な形式

filename attribute control-val test-val [*attribute control-val test-val*]*

filename	デフォルト形式の *filename* と同じです。
attribute control-val test-val	各ファイルの制御目録とテスト目録間で異なるファイル属性の説明。各エントリには、各目録の属性値が含まれます。属性の定義については、`bart_manifest(4)` を参照してください。

プログラム的な出力の各行には、1 つのファイルに含まれるすべての属性の相違が示されます。

プログラム的な次の出力は、`/etc/passwd` ファイルの属性の相違を示しています。この出力から、`size`、`mtime`、および `contents` 属性に変化があったことがわかります。

```
/etc/passwd size 74 81 mtime 3c165879 3c165979
contents daca28ae0de97afd7a6b91fde8d57afa 84b2b32c4165887355317207b48a6ec7
```

終了ステータス

目録生成ツール

目録生成ツールは、次の終了値を返します。

`0`	成功
`1`	ファイル処理時の致命的でないエラー (アクセス権の問題など)
`>1`	致命的エラー (無効なコマンド行オプションなど)

レポートツール

レポートツールは、次の終了値を返します。

`0`	相違は報告されませんでした
`1`	相違が見つかりました
`>1`	比較実行中の致命的エラー

使用例

例 1 チェックサムを計算せずにデフォルトの目録を作成する

次のコマンド行は、`/` ファイルシステムに含まれるすべてのファイルで構成されるデフォルトの目録を作成します。`-n` オプションを指定すると、チェックサムが計算されないため、目録がより速く生成されます。

```
bart create -n
```

例 2 指定したサブツリーの目録を作成する

次のコマンド行は、`/home/nickiso` サブツリーに含まれるすべてのファイルで構成される目録を作成します。

```
bart create -R /home/nickiso
```

例 3　標準入力を使って目録を作成する

次のコマンド行は、find(1) コマンドの出力を使って、カタログ化するファイルのリストを生成します。-I オプションを指定した bart create コマンドの入力として、find の出力が使用されています。

```
find /home/nickiso -print | bart create -I
```

例 4　規則ファイルを使って目録を作成する

次のコマンド行は、rules という規則ファイルを使って、カタログ化するファイルを指定します。

```
bart create -r rules
```

例 5　2 つの目録を比較し、プログラム的な出力を生成する

次のコマンド行は、2 つの目録を比較し、プログラムで解析するのに適した出力を生成します。

```
bart compare -p manifest1 manifest2
```

例 6　2 つの目録を比較し、無視する属性を指定する

次のコマンド行は、2 つの目録を比較します。dirmtime、lnmtime、および mtime 属性は比較されません。

```
bart compare -i dirmtime,lnmtime,mtime manifest1 manifest2
```

例 7　規則ファイルを使って 2 つの目録を比較する

次のコマンド行は、rules という規則ファイルを使って 2 つの目録を比較します。

```
bart compare -r rules manifest1 manifest2
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	security/bart
インタフェースの安定性	確実

関連項目

cksum(1), digest(1), find(1), bart_manifest(4), bart_rules(4), attributes(5)

注意事項

特定のシステムライブラリのファイル属性は、システムのブート時に一時的に変更されることがあります。誤った警告が発生しないようにするには、2 つの目録が作成されたときのシステムの状態が同じ場合、つまり、どちらもシングルユーザーモードで作成された場合か、どちらもマルチユーザーモードで作成された場合にのみ、目録を比較するようにしてください。

名前

beadm – ZFS ブート環境の管理

形式

```
beadm create [-a] [-d description]
      [-e non-activeBeName | beName@snapshot]
      [-o property=value] ... [-p zpool] beName

beadm create beName@snapshot

beadm destroy [-fF] beName | beName@snapshot

beadm list [-a | -ds] [-H] [beName]

beadm mount beName mountpoint

beadm unmount [-f] beName

beadm rename beName newBeName

beadm activate beName
```

機能説明

beadm コマンドは、ZFS ブート環境 (BE) を管理するためのユーザーインタフェースです。このユーティリティーは、単一のシステムで複数の Oracle Solaris インスタンスを管理するシステム管理者によって使用されます。

beadm を使用すると、次の操作を行うことができます。

- アクティブな BE に基づいた、新しい BE の作成
- アクティブでない BE に基づいた、新しい BE の作成
- 既存の BE のスナップショットの作成
- 既存のスナップショットに基づいた、新しい BE の作成
- 新しい BE の作成と別の zpool へのコピー
- 既存のアクティブでない BE のアクティブ化
- BE のマウント
- BE のマウント解除
- BE の破棄
- BE のスナップショットの破棄
- 既存のアクティブでない BE の名前変更
- スナップショットとデータセットに関する情報の表示

サブコマンド

beadm コマンドには、下記のようなサブコマンドおよびオプションがあります。これらのサブコマンドおよびオプションの多くの使用法については、次の「使用例」で説明しています。

beadm *(引数なし)*
: コマンドの使用法を表示します。

beadm create [-a] [-d *description*] [-e *non-activeBeName* | *beName@snapshot*] [-o *property=value*] ... [-p *zpool*] *beName*
: *beName* という新しいブート環境を作成します。-e オプションを指定しない場合、新しいブート環境は現在実行中のブート環境のクローンとして作成されます。-d オプションを指定する場合、記述が x86 システムの GRUB メニューまたは SPARC システムのブートメニューにある BE エントリのタイトルとしても使用さ

れます。-d オプションを指定しない場合、*beName* がタイトルとして使用されます。入れ子の BE では、-p オプションの使用はサポートされません。また、ブート不可能な入れ子の BE と、ブート不可能な入れ子の BE のスナップショットは、-e オプションとともに使用できません。

-a
作成と同時に、新しく作成された BE をアクティブにします。デフォルトでは、新しく作成された BE をアクティブにしません。

-d *description*
関連付けられた記述を含む新しい BE を作成します。

-e *non-activeBeName*
既存のアクティブでない BE から新しい BE を作成します。入れ子の BE では、ブート可能な BE のみがこのオプションとともに使用できます。

-e *beName@snapshot*
beName という BE の既存のスナップショットから新しい BE を作成します。入れ子の BE では、ブート可能な BE のスナップショットのみがこのオプションとともに使用できます。

-o *property=value*
特定の ZFS プロパティーを使って新しい BE のデータセットを作成します。複数の -o オプションを指定できます。-o オプションの詳細は、zfs(1M) を参照してください。

-p *zpool*
指定された *zpool* に新しい BE を作成します。このオプションを指定しない場合、デフォルトの動作では、元の BE と同じプールに新しい BE を作成します。このオプションは、入れ子の BE の内側ではサポートされません。

beadm create *beName@snapshot*
beName という既存の BE のスナップショットを作成します。入れ子の BE の内側では、ブート可能な BE のスナップショットのみを作成できます。入れ子の BE の内側にいるときは、ブート可能な BE またはブート可能ではないがリブート時にアクティブとしてマークされない BE のみ破棄できます。

beadm destroy [-fF] *beName* | *beName@snapshot*
beName というブート環境を破棄するか、*beName@snapshot* というブート環境の既存のスナップショットを破棄します。ブート環境を破棄すると、そのブート環境のスナップショットもすべて破棄されます。このコマンドを使用するときは、注意が必要です。

-f
ブート環境が現在マウントされている場合、強制的にマウント解除します。

-F
ブート環境を破棄するための確認を求めずに操作を強制的に実行します。

`beadm list` [`-a` | `-ds`] [`-H`] [*beName*]

beName という既存のブート環境に関する情報を一覧表示します。*beName* を指定しない場合は、すべてのブート環境に関する情報を一覧表示します。`Active` フィールドは、ブート環境が現在アクティブであるか (`N` で表現)、リブート時にアクティブになるか (`R` で表現)、またはその両方であるか (`NR` で表現) を示します。入れ子の BE の内側にあるブート不可能な BE は、感嘆符 (`!`) で表現されます。

機械解析可能な出力の各行には、1 番目のフィールドとしてブート環境名が含まれます。`Space` フィールドはバイト単位で表示され、`Created` フィールドは UTC 形式で表示されます。その他のオプションを付けないで `-H` オプションを使用すると、2 番目のフィールドにブート環境の UUID が表示されます。ブート環境に UUID がない場合、このフィールドは空白になります。「使用例」の節を参照してください。入れ子の BE の内側では、UUID フィールドは実際には入れ子の BE に関連付けられている親 ID を表します。

`-a`

ブート環境に関する利用可能なすべての情報を一覧表示します。この情報には、従属ファイルシステムおよびスナップショットが含まれます。

`-d`

ブート環境に属しているすべての従属ファイルシステムに関する情報を一覧表示します。

`-s`

ブート環境のスナップショットに関する情報を一覧表示します。

`-H`

ヘッダー情報を一覧表示しません。一覧情報の各フィールドは、セミコロンで区切られます。

`beadm mount` *beName mountpoint*

マウントポイントで *beName* というブート環境をマウントします。マウントポイントは、既存の空ディレクトリである必要があります。

`beadm unmount` [`-f`] *beName*

beName というブート環境をマウント解除します。

`-f`

ブート環境が現在ビジー状態であっても強制的にマウント解除します。

`beadm rename` *beName newBeName*

beName というブート環境の名前を *newBeName* に変更します。入れ子の BE では、ブート可能な BE の名前変更のみを行えます。

`beadm activate` *beName*

次回のリブート時に *beName* をアクティブな BE にします。入れ子の BE では、ブート可能な BE のみをアクティブ化できます。

入れ子の BE のサポート

beadm は、入れ子の BE の概念を、厳密にはそれが非大域ゾーンの BE に属している場合にサポートします。現時点では、beadm は非大域ゾーンの内側からのみ入れ子の BE を管理できます。

beadm は、いくつかの例外を除き、大域ゾーンから行うのとほとんど同様に非大域ゾーンの内側で機能します。まず、beadm create の -p (代替プール) オプションは非大域ゾーン内ではサポートされていません。次に、指定された入れ子の BE (または BE のスナップショット) に対し、ブート可能かブート可能でないかを判断するための区別がつけられます。入れ子の BE がブート可能であるのは、それが現在アクティブになっている大域ゾーンの BE に関連付けられている (つまり、アクティブな大域ゾーンの BE の UUID と同じ親 ID を共有する) 場合です。それ以外の場合、それはブート可能ではなく、beadm list のアクティブ列に「!」のマークが付きます。注: 非大域ゾーンの管理者は beadm activate を使ってそのような BE をアクティブとしてマークできますが、非大域ゾーンをリブートしても BE はロードされません。これは、その BE が非アクティブな大域ゾーンの BE に関連付けられているからです。そのため、これらの条件に基づいて、beadm はブート不可能な BE でいくつかの動作を制限します。

- ブート不可能でかつ、リブート時にアクティブとしてマークされる入れ子の BE を破棄できません。
- ブート不可能な BE をアクティブ化することはできません。
- ブート不可能な BE のスナップショットを作成することはできません。
- beadm create の -e オプションでブート不可能な BE または BE のスナップショットを使用することはできません。
- ブート不可能な BE の名前変更を行うことはできません。

使用例

例 1　アクティブな BE を使用した新しい BE の作成

次のコマンドは、現在の BE のクローンを作成して、BE1 という新しい BE を作成します。

```
# beadm create BE1
```

例 2　アクティブでない BE を使用した新しい BE の作成

次のコマンドは、BE1 という既存のアクティブでない BE のクローンを作成して、BE2 という新しい BE を作成します。

```
# beadm beadm create -e BE1 BE2
```

例 3　既存の BE のスナップショットの作成

次のコマンドは、BE1 という既存の BE の now というスナップショットを作成します。

```
# beadm create BE1@now
```

例 4　スナップショットのクローンの作成による、新しい BE の作成

次のコマンドは、BE1 の既存のスナップショットのクローンを作成して、BE3 という新しい BE を作成します。

```
# beadm create -e BE1@now BE3
```

例 5　指定された zpool の新しい BE の作成

次のコマンドは、現在実行中の BE に基づいて、BE4 という新しい BE を作成します。このコマンドは、rpool2 という zpool に新しい BE を作成します。

```
# beadm create -p rpool2 BE4
```

例 6　圧縮を有効にした状態で指定された zpool の新しい BE の作成

次のコマンドは、現在実行中の BE に基づいて、BE5 という新しい BE を作成します。このコマンドは、rpool2 という zpool に新しい BE を作成して、圧縮を有効にした状態でそのデータセットを作成します。

```
# beadm create -p rpool2 -o compression=on BE5
```

例 7　新しい BE の作成と説明の提供

次のコマンドは、現在実行中の BE に基づいて BE6 という新しい BE を作成し、その説明を提供します。

```
# beadm create -d "BE6 used as test environment" BE6
```

例 8　BE の有効化

次のコマンドは、BE3 という既存の有効でない BE を有効にします。

```
# beadm activate BE3
```

例 9　BE のマウント

次のコマンドは、BE3 という BE を /mnt にマウントします。

```
# beadm mount BE3 /mnt
```

例 10　BE のマウント解除

次のコマンドは、BE3 という BE をマウント解除します。

```
# beadm unmount BE3
```

例 11　BE の破棄

次のコマンドは、確認を求めずに BE3 という BE を破棄します。

```
# beadm destroy -F BE3
```

例 12 スナップショットの破棄

次のコマンドは、BE1 の now というスナップショットを破棄します。

```
# beadm destroy BE1@now
```

例 13 BE の名前の変更

次のコマンドは、BE1 という既存の有効でない BE の名前を BE3 に変更します。

```
# beadm rename BE1 BE3
```

例 14 すべての BE の一覧表示

次のコマンドは、既存の BE をすべて一覧表示します。

```
# beadm list
BE  Active Mountpoint Space  Policy Created
--  ------ ---------- -----  ------ -------
BE2 -      -          72.0K  static 2008-05-21 12:26
BE3 -      -          332.0K static 2008-08-26 10:28
BE4 -      -          15.78M static 2008-09-05 18:20
BE5 NR     /          7.25G  static 2008-09-09 16:53
```

例 15 データセットおよびスナップショット情報を含むすべての BE の一覧表示

次のコマンドは、既存の BE をすべて一覧表示し、それらのブート環境に関するデータセットおよびスナップショット情報をすべて一覧表示します。

```
# beadm list -d -s
BE/Dataset/Snapshot     Active Mountpoint Space   Policy Created
-------------------     ------ ---------- -----   ------ -------
BE2
   p/ROOT/BE2           -      -          36.0K   static 2008-05-21 12:26
   p/ROOT/BE2/opt       -      -          18.0K   static 2008-05-21 16:26
   p/ROOT/BE2/opt@now   -      -          0       static 2008-09-08 22:43
   p/ROOT/BE2@now       -      -          0       static 2008-09-08 22:43
BE3
   p/ROOT/BE3           -      -          192.0K  static 2008-08-26 10:28
   p/ROOT/BE3/opt       -      -          86.0K   static 2008-08-26 10:28
   p/ROOT/BE3/opt/local -      -          36.0K   static 2008-08-28 10:58
BE4
   p/ROOT/BE4           -      -          15.78M  static 2008-09-05 18:20
BE5
   p/ROOT/BE5           NR     /          6.10G   static 2008-09-09 16:53
   p/ROOT/BE5/opt       -      /opt       24.55M  static 2008-09-09 16:53
   p/ROOT/BE5/opt@bar   -      -          18.38M  static 2008-09-10 00:59
   p/ROOT/BE5/opt@foo   -      -          18.38M  static 2008-06-10 16:37
   p/ROOT/BE5@bar       -      -          139.44M static 2008-09-10 00:59
   p/ROOT/BE5@foo       -      -          912.85M static 2008-06-10 16:37
```

例 16 BE に関するデータセットおよびスナップショット情報の一覧表示

次のコマンドは、BE5 に関するデータセットおよびスナップショット情報をすべて一覧表示します。

```
# beadm list -a BE5
BE/Dataset/Snapshot   Active Mountpoint Space   Policy Created
-------------------   ------ ---------- -----   ------ -------
BE5
   p/ROOT/BE5         NR     /          6.10G   static 2008-09-09 16:53
   p/ROOT/BE5/opt     -      /opt       24.55M  static 2008-09-09 16:53
   p/ROOT/BE5/opt@bar -      -          18.38M  static 2008-09-10 00:59
   p/ROOT/BE5/opt@foo -      -          18.38M  static 2008-06-10 16:37
   p/ROOT/BE5@bar     -      -          139.44M static 2008-09-10 00:59
   p/ROOT/BE5@foo     -      -          912.85M static 2008-06-10 16:37
```

例 17 機械解析可能な形式での一覧表示

次のコマンドは、すべての BE に関する情報を機械解析可能な形式で一覧表示します。

```
# beadm list -H
BE2;;;;55296;static;1211397974
BE3;;;;339968;static;1219771706
BE4;;;;16541696;static;1220664051
BE5;215b8387-4968-627c-d2d0-f4a011414bab;NR;/;7786206208;static;1221004384
```

例 18 ブート不可能な BE の表示

次のコマンドは、すべての BE を一覧表示します。非大域ゾーンの内側で実行している場合、ブート可能な BE とブート不可能な BE の両方が表示されます。ブート可能でない BE は、アクティブ列の感嘆符 (!) で示されます。

```
# beadm list
BE    Active Mountpoint Space   Policy Created
--    ------ ---------- -----   ------ -------
zbe-0 -      -          29.22M  static 2011-03-04 09:14
zbe-1 NR     /          815.10M static 2011-03-04 09:28
zbe-2 -      -          35.0K   static 2011-03-04 09:28
zbe-3 -      -          35.0K   static 2011-03-04 09:28
zbe-4 -      -          35.0K   static 2011-03-04 09:28
zbe-5 !      -          35.0K   static 2011-03-04 11:47
zbe-6 !      -          54.0K   static 2011-03-07 14:37
```

終了ステータス

0
成功。

>0
失敗。

ファイル

/var/log/beadm/*beName*/create.log.*yyyymmdd_hhmmss*

beadm create 出力を取得するために使用されるログ。ファイル名の時刻指定部分については、次に説明します。

- *yyyymmdd_hhmmss* (例: 20071130_140558)。
- *yyyy* (2007 年)
- *mm* (11 月)
- *dd* (30 日)
- *hh* (14 時)
- *mm* (05 分)
- *ss* (58 秒)

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/boot-environment-utilities
インタフェースの安定性	確実

関連項目

zfs(1M), attributes(5)

名前

boot – システムカーネルまたはスタンドアロンプログラムのブート

形式

SPARC

```
boot [OBP names] [file] [-aLV] [-F object] [-D default-file]
     [-Z dataset] [boot-flags] [—] [client-program-args]
```

x86

```
kernel$ /platform/i86pc/kernel/$ISADIR/unix [boot-args]
     [-B prop=val [,val...]]
```

機能説明

ブートストラップとは、スタンドアロンプログラムを読み込んで実行する処理のことです。ここでのブートストラップとは、ブート可能なオペレーティングシステムを読み込んで実行することを意味します。通常、スタンドアロンプログラムはオペレーティングシステムカーネル (kernel(1M) を参照) ですが、代わりに任意のスタンドアロンプログラムをブートすることもできます。SPARC ベースのシステムでは、オペレーティングシステム以外にブートできるスタンドアロンプログラムの代表的な例として、マシンの診断モニターがあります。

スタンドアロンプログラムが動的にリンクされる実行可能プログラムとして識別された場合、boot は実行可能形式によって指定されたインタプリタ (リンカー/ローダー) をロードし、そのインタプリタに制御を渡します。スタンドアロンプログラムが静的にリンクされている場合は、そのスタンドアロンプログラムに直接ジャンプします。

カーネルは読み込まれると、UNIX システムを起動して、必要なファイルシステム (vfstab(4) を参照) をマウントし、/usr/sbin/init を実行して、システムを /etc/inittab で指定されている「initdefault」状態にします。inittab(4) を参照してください。

SPARC でのブートストラップ手続き

SPARC ベースのシステムでは、ほとんどのマシンでのブートストラップ手続きは次の基本フェーズからなります。

マシンの電源を投入すると、(PROM 内の) システムファームウェアが電源投入時自己診断テスト (POST) を実行します。この診断テストの形式と範囲は、システムに搭載されているファームウェアのバージョンによって異なります。

診断テストが正常に完了した後、ファームウェアが使用する不揮発性記憶領域に適切なフラグが設定されていれば、ファームウェアは自動ブートを試みます。ファームウェアは、読み込むファイルの名前とそのファイルを読み込むデバイスを選択することができます。

これらのフラグと名前は、シェルから eeprom(1M) コマンドを使用するか、またはシステムの停止後に ok プロンプトで PROM コマンドを使用することにより、設定できます。

第 2 レベルのプログラムは、ファイルシステム固有のブートブロック (ディスクからブートする場合) または inetboot または wanboot (ネットワークからブートする場合) のどれかです。

ネットワークブート

ネットワークブートは 2 段階で行われます。 まず、クライアントが、二次ブートプログラムの読み込みに必要な IP アドレスとその他のすべてのパラメータを取得します。次に、二次ブートプログラムがブートデバイスからブートアーカイブをロードします。

IP アドレスは、 PROM で使用可能な機能と PROM の構成に応じて、RARP、DHCP、手動構成のいずれかの方法を使って取得できます。sun4u および sun4v カーネルアーキテクチャーのマシンは、DHCP に対応した PROM を備えています。

次に、2 種類のネットワークブート方式を指定する boot コマンド構文を示します。

```
boot net:rarp
boot net:dhcp
```

次のコマンドを見てください。

```
boot net
```

このコマンドのように、rarp または dhcp 指示子を指定しないと、net が別名になっているネットワークインタフェースを介して、デフォルトのネットワークブート方式が呼び出されます。

以降では、RARP/bootparams を使用するネットワークブートのイベントシーケンスについて説明します。そのあとで、DHCP を用いたシーケンスについて説明します。

RARP/bootparams を使ってネットワークからブートする場合、PROM はまず、応答を受信するまで逆 ARP 要求をブロードキャストします。応答を受信すると、TFTP 要求をブロードキャストして inetboot の最初のブロックを取得します。続いて、その最初のブロック要求に一番先に応答したサーバーに後続の要求が送信されます。読み込みの完了後、inetboot も同様に、逆 ARP を使って IP アドレスを取得した後、bootparams RPC 呼び出し (bootparams(4) を参照) をブロードキャストすることで、構成情報とルートファイルシステムを検索します。次に、inetboot が NFS を使用してブートアーカイブをロードし、そのアーカイブに制御を渡します。

DHCP を使用してネットワークからブートする場合、PROM はハードウェアアドレスおよびカーネルアーキテクチャをブロードキャストし、IPアドレス、ブートパラメータ、およびネットワーク構成情報を要求します。(可能性のある複数のサーバーの中から) DHCP サーバーが応答し選択されると、そのサーバーがクライアントに IP アドレスおよびクライアントのブートに必要な他のすべての情報を送信します。この情報を受信したクライアントの PROM は、読み込むファイルの名前を調べ、そのファイル名が HTTP URL であるかどうかにより、2 種類の動作のいずれかを実行します。ファイル名が HTTP URL でなかった場合、PROM は、inetboot をダウンロードし、それをメモリー内に読み込んで実行します。inetboot はブートアーカイブをロードし、そのブートアーカイブがマシンを引き継いで inetboot を

解放します。その後、開始スクリプトが DHCP エージェント (dhcpagent(1M) を参照) を開始します。以降の DHCP アクティビティは、その DHCP エージェントによって実行されます。

読み込むファイルが HTTP URL の場合、PROM は、その参照ファイルを HTTP を使って読み込みます。クライアントが HMACSHA-1 鍵を使って構成されていた場合、クライアントは読み込んだファイルの整合性を検査した後で、そのファイルを実行します。そのファイルは wanboot バイナリであるとみなされます。WAN ブートプロセスは、DHCP または NVRAM プロパティーのどちらかを使用してインストールサーバーとルーター、およびその接続に必要なプロキシを発見するように構成できます。ブートされた wanboot は、処理を継続できるだけの情報がそろっているかどうかを判断します。必要な情報が不足していた場合は、適切なエラーを出力して処理を終了するか、コマンドインタプリタを起動して必要な構成情報の入力をユーザーに求めます。必要な情報を取得した wanboot は、HTTP 経由でブートローダーをメモリー内に読み込みます。クライアントに暗号化鍵がインストールされている場合、wanboot は、ブートローダーの署名とそれに付随するハッシュを確認します。暗号化鍵は存在するが、ハッシュキーがない場合はエラーとなります。

wanboot ブートローダーは、HTTP またはセキュリティー保護された HTTP のどちらかを使用してクライアントと通信できます。 HTTP を使用する場合でかつ、クライアントが HMAC SHA-1 鍵を使って構成されていた場合、ブートローダーはルートファイルシステムの整合性検査を実行します。ルートファイルシステムがメモリーに読み込まれた (場合によっては、さらに整合性検査が実行された) あと、ブートアーカイブがサーバーから転送されます。なお、wanboot.conf(4)ファイルで boot_logger URL が指定されていた場合、wanboot は処理状況に関するログを定期的に記録します。

すべての PROM が URL を処理できるわけではありません。クライアントにその機能が備わっているかどうかを判断するには、list-security-keys OBP コマンド (monitor(1M) を参照) を使用します。

現在のところ、x86 プラットフォーム上では WANブートは利用できません。

wanboot コマンド行

クライアントプログラムが wanboot である場合、クライアントプログラムは、次の形式の client-program-args を受け入れます。

```
boot ... -o opt1[,opt2[,...]]
```

ここで、各オプションに指定できるアクションは、次のとおりです。

dhcp
: DHCP 経由で構成パラメータを取得するように wanboot に指示します。

prompt
: コマンドインタプリタをブートするように wanboot に指示します。

<cmd>
下記のインタプリタコマンドのいずれか。

...または、下のインタプリタパラメータ名を使用した代入文。

`wanboot` コマンドインタプリタ

ブート時に `client-program-args` として「`-o prompt`」を指定すると、`wanboot` コマンドインタプリタがブートされます。単一のコマンドまたは代入文、もしくは、コンマで区切られた一連のコマンドまたは代入文が入力可能です。利用可能な構成パラメータは、次のとおりです。

`host-ip`
クライアントの IP アドレス (ドット区切り 10 進表記)。

`router-ip`
デフォルトのルーターの IP アドレス (ドット区切り 10 進表記)。

`subnet-mask`
サブネットマスク (ドット区切り 10 進表記)。

`client-id`
DHCP クライアント識別子 (引用符付き ASCII 文字列または 16 進 ASCII)。

`hostname`
DHCP トランザクションで要求するホスト名 (ASCII)。

`http-proxy`
HTTP プロキシサーバー指定 (IPADDR[: PORT])。

利用可能な鍵名は、次のとおりです。

`3des`
トリプル DES 暗号鍵 (48 個の 16 進 ASCII 文字)。

`aes`
AES 暗号鍵 (32 個の 16 進 ASCII 文字)。

`sha1`
HMAC SHA-1 署名鍵 (40 個の 16 進 ASCII 文字)。

最後に、WANブート CGI の URL を参照する方法を示します。

`bootserver`
WANブート CGI の URL (OBP の `file` パラメータと同等)。

インタプリタが受け入れるコマンドは、次のとおりです。

`help`
利用可能なコマンドの簡単な説明文を出力します。

var=*val*
: *val*を *var* に代入します。ここで、*var*は、構成パラメータ名、鍵名、`bootserver` のいずれかです。

var=
: パラメータ *var* の設定を解除します。

`list`
: すべてのパラメータとその設定値を一覧表示します (OBP 経由で取得された鍵の値は表示されません)。

`prompt`
: 未設定パラメータに対する値の入力を、ユーザーに求めます。各パラメータの名前と現在値 (もしあれば) が出力されます。ユーザーは、Return キーを押して現在値を受け入れることもできますし、新しい値を入力することもできます。

`go`
: すべての値の入力を完了したら、インタプリタを終了しブート処理を継続します。

`exit`
: ブートインタプリタを終了し、OBPの `ok` プロンプトに戻ります。

前述の代入文やコマンドは、コマンド行の `-o` オプションの一部として、いくつでも指定できますが、boot コマンドの引数の長さには 128 バイトの OBP 制限があります。たとえば、「`-o list,go`」とすると、パラメータの現在値 (デフォルト値) が一覧表示された後、ブート処理が継続します。

iSCSI ブート

iSCSI ブートは、x86 と SPARC の両方でサポートされています。

x86 上の iSCSI ブート

x86 上の iSCSI ブートの場合は、ブートされるホストに iBFT (iSCSI Boot Firmware Table) に対応した NIC が装着されているか、またはメインボードの BIOS が iBFT に対応している必要があります。ACPI (Advanced Configuration and Power Interface) 3.0b 仕様で定義された iBFT では、iSCSI ブートプロセスに有効な、さまざまなパラメータを含む情報のブロックが指定されています。

ファームウェアで実装された iBFT は、iSCSI ターゲットへの接続を確立することによって、ブート可能なデバイスとしてブート中に BIOS 内に iSCSI ディスクを提供します。iSCSI ブートの残りの処理は、ローカルディスクからのブートと同じです。

ユーザーが iBFT を正しく構成するには、各ハードウェアベンダーのドキュメントを参照する必要があります。

SPARC 上の iSCSI ブート

SPARC 上の iSCSI ブートは OpenBoot レベル 4.31 以降でサポートされ、特定の NIC は必要ありません。

OpenBoot でのブートコマンドには、*keyword=value* 形式に従った、宛先の iSCSI ターゲットを識別するための一連のキーワードを指定します。iSCSI ブートコマンドの完全な形式は次のとおりです。

```
boot net:iscsi-target-ip=t-ip,iscsi-target-name=name
     host-ip=h-ip[,router-ip=r-ip]
     [,subnet-mask=m-ip]
     [,iscsi-port=port]
     [,iscsi-lun=lun]
     [,iscsi-partition=partition]
```

前述のキーワードの説明を次に示します。

`host-ip`	ブート対象ホストの IP アドレス。
`router-ip`	経路指定するゲートウェイの IP アドレス。
`subnet-mask`	`host-ip` のサブネットマスク。
`iscsi-target-ip`	OS を格納している iSCSI ターゲットの IP アドレス。
`iscsi-target-name`	OS を格納している iSCSI ターゲットの名前。
`iscsi-partition`	ブート可能なルートを含むパーティション。
`iscsi-port`	ターゲットの IP ポート。
`iscsi-lun`	ターゲット上のブートされる LUN。

`iscsi-target-ip`、`route-ip`、および `subnet-mask` の値は、標準の IPv4 ドット区切り 10 進数の形式です。たとえば、`subnet-mask` に `255.255.255.0` を指定します。IPv6 は、現在の OpenBoot 実装ではサポートされていません。

`iscsi-port` の値は、1 から 65535 までの 10 進数です。

`iscsi-lun` の値は、ダッシュで区切られた 16 進数の LUN、*ffff-ffff-ffff-ffff* の形式です。詳細は、RFC 4173 のセクション 5 を参照してください。先頭のゼロと以降のダッシュは除外できます。そのため、たとえば、`3` は `0003-0000-0000-0000` と同等です。

`iscsi-partition` の値は、ルートパーティションを指定するために使用される 1 つの ASCII 文字です。一般には、`a` です。

`iscsi-target-name` の値は、RFC 3720 と RFC 3722 によって指定されている文字列の形式です。

ターゲット側での CHAP 認証を提供するために、2 つのセキュリティーキーが追加されます。これらのキーは次のとおりです。

`chap-user`	CHAP 名
`chap-password`	CHAP シークレット

現在、これらの 2 つのキーは、OpenBoot PROM (ok) プロンプトでコマンド set-ascii-security-key を使用して設定できます。例:

```
ok set-ascii-security-key chap-user chap name
ok set-ascii-security-key chap-password chap password
```

双方向の認証は、まだサポートされていません。これらの 2 つの変数は、OpenBoot PROM プロンプトでのみ変更できます。

DHCP サーバーから iSCSI ブート情報を取得するために、RFC 4173 がサポートされています。DHCP サーバーは、ブート対象クライアントのルートパスオプションを指定する必要があります。そのあと、クライアントは、単純コマンドを使用して iSCSI ブートを実行できます。

```
boot net:dhcp
```

現在は、iSCSI ブート中に物理ブートデバイスのパスを取得するために、キー boot-device が使用されます。このキーは、Solaris のインストール中に設定されます。キー値を手動で変更すると、iSCSI ブートが停止することがあります。

ディスクからのブート

ディスクからブートする場合、OpenBoot PROM ファームウェアは、ブートデバイスとして指定されたパーティションのブロック 1 ～ 15 からブートブロックを読み取ります。このスタンドアロンのブートプログラムには通常、ブートアーカイブを読み取ることができる、ファイルシステム固有のリーダーが含まれています。

スタンドアロンプログラムのパス名が相対パス名の場合 (スラッシュで始まらない場合)、第 2 レベルのブートにおいて、プラットフォームに依存する検索パスでスタンドアロンプログラムが検索されます。このパスには必ず、/platform/*platform-name* が含まれています。多くのSPARC プラットフォームでは、次にプラットフォーム固有のパスエントリ /platform/*hardware-class-name* が検索されます。filesystem(5) を参照してください。絶対パス名の場合、boot は指定されたパスを使用します。その後、boot プログラムは該当するアドレスからスタンドアロンプログラムをロードし、制御を渡します。

ブートアーカイブがブートデバイスから転送されたあと、Solaris はマシンを初期化し、その制御を引き継ぐことができます。このプロセスは、下の「ブートアーカイブのフェーズ」でさらに詳細に説明されており、すべてのプラットフォーム上で同一です。

ファイル名がコマンド行で、または boot-file NVRAM 変数などで指定されていない場合、boot はシステムにインストールされているソフトウェア、ハードウェアおよびファームウェアの能力に基づいて、読み込むべきデフォルトのファイルを選択します。

カーネルへのパスには、スペースを含めてはいけません。

ZFS からのブート

ZFS では、デバイス指定子によって単一のルートファイルシステムではなく、ストレージプールが識別されるため、ZFS からのブートは UFS からのブートとは異なります。ストレージプールには、複数のブート可能データセット (つまり、ルートファイルシステム) が含まれていることがあります。そのため、ZFS からブートする場合、ブートデバイスを指定するだけでは十分ではありません。ブートデバイスによって識別されたプール内のルートファイルシステムも指定する必要があります。デフォルトでは、プールの `bootfs` プロパティーで指定されているデータセットが、ブート用に選択されます。このデフォルトの選択は、`-Z` オプションで代替のブート可能データセットを指定することによって上書きできます。ZFS プール内のブート可能データセットを一覧表示するには、`-L` オプションを使用します。

ブートアーカイブのフェーズ

ブートアーカイブには、メモリー内ディスクを使用してマウントされるファイルシステムイメージが含まれています。このイメージは自己記述型であり、特にブートブロックにはファイルシステムリーダーが含まれています。このファイルシステムリーダーは RAM ディスクイメージをマウントして開いたあと、その中に含まれているカーネルを読み取って実行します。 デフォルトでは、このカーネルは次の場所にあります。

```
/platform/‘uname -i‘/kernel/unix
```

ZFS からブートする場合は、アーカイブとカーネルファイルの両方のパス名が、前の節で説明した、ブート用に選択されたルートファイルシステム (つまり、データセット) 内で解決されます。

カーネルの初期化は、I/O をオンにし、ルートファイルシステムをマウントできるまで、メモリー内ファイルシステムから必要なドライバとモジュールを読み込むことにより続行されます。ルートファイルシステムがマウントされると、メモリー内ファイルシステムは必要なくなり、破棄されます。

OpenBoot PROM boot コマンドの動作

OpenBoot `boot` コマンドは、次の形式の引数をとります。

`ok boot [`*device-specifier*`] [`*arguments*`]`

デフォルトの `boot` コマンドに引数はありません。

```
ok boot
```

`boot` コマンド行で *device-specifier* を指定しない場合、OpenBoot は通常、*boot-device* 変数または *diag-device* `NVRAM` 変数を使用します。オプションの *arguments* をコマンド行で指定しないと、OpenBoot は通常、*boot-file* または *diag-file* `NVRAM` 変数をデフォルトの `boot` 引数として使用します。(システムが診断モードの場合、*boot-device* および *boot-file* の代わりに、*diag-device* と *diag-file* が使用されます)。

arguments には複数の文字列を指定できます。すべての *argument* 文字列は二次ブートプログラムに渡され、OpenBoot では解釈されません。

boot コマンド行に *arguments* を指定した場合、*boot-file* または *diag-file* NVRAM 変数のどちらも使用されません。NVRAM 変数の内容がコマンド行の引数とマージされることもありません。次に例を示します。

```
ok boot -s
```

このコマンドでは、*boot-file* および *diag-file* の設定値は無視され、文字列 -s は *arguments* として解釈されます。この場合、boot は *boot-file* または *diag-file* の内容を使用しません。

以前の PROM におけるコマンド例を示します。

```
ok boot net
```

このコマンドでは引数が指定されていないため、代わりに、*boot-file* または *diag-file* の設定値 (設定されている場合) が、boot に渡すデフォルトのファイル名および引数として使用されます。ほとんどの場合、システムタイプ、システムのハードウェアとファームウェア、およびルートファイルシステムにインストールされているソフトウェアに基づいて boot コマンドに適切なデフォルト値を選択させるのが、最善の方法です。*boot-file* または *diag-file* を変更すると、状況によっては予期せぬ結果を招くおそれがあります。

これは、大部分の OpenBoot 2.x および 3.x ベースのシステムにおける一般的な動作です。ただし、プラットフォームによっては動作が異なる可能性もあります。

次のコマンドを見てください。

ok **boot cdrom**

...このコマンドも通常は、引数を指定しません。したがって、*boot-file* に 64ビットカーネルのファイル名が設定されている環境で、「boot cdrom」と入力してインストール CD または DVD をブートしようとしても、インストールメディアに 32 ビットカーネルしか含まれていなければ、そのブート処理は失敗します。

使用される boot コマンドの形式によっては、*boot-file*または *diag-file* の内容が無視されるので、プロダクションシステムで *boot-file* に依存することは一般に推奨しません。

ローカル (CD または DVD 上) に存在する wanboot コピーから WAN ブートを実行するには、次のコマンドを使用する必要があります。

ok **boot cdrom -F wanboot - install**

最近の PROM に含まれるネットワークブートサポートパッケージは、次の構文をサポートするように強化されており、引数を処理できるようになっています。

[*protocol*,] [*key*=*value*,]*

すべての引数は省略可能であり、指定する順序にも制限はありません。ただし、リストの末尾でない限り、引数の後にはコンマが必要です。ここで指定した引数は、すべてのデフォルト値よりも優先されるほか、DHCP を使ってブートする場合には、それらの引数に対応する DHCP サーバーから提供される構成情報よりも優先されます。

前述の *protocol* には、使用するアドレス検索プロトコルを指定します。

key=value 属性ペアには、以下の構成パラメータを指定します。

`tftp-server`
: TFTP サーバーの IP アドレス。

`file`
: TFTP を使ってダウンロードするファイルの名前、または WANブートの URL。

`host-ip`
: クライアントの IP アドレス (ドット区切り 10 進表記)。

`router-ip`
: デフォルトのルーターの IP アドレス。

`subnet-mask`
: サブネットマスク (ドット区切り 10 進表記)。

`client-id`
: DHCP クライアント識別子。

`hostname`
: DHCP トランザクションで使用するホスト名。

`http-proxy`
: HTTP プロキシサーバー指定 (IPADDR[: PORT])。

`tftp-retries`
: TFTP の最大リトライ回数。

`dhcp-retries`
: DHCP の最大リトライ回数。

ネットワークブートサポートパッケージによって処理される一連の引数は、次のいずれかの方法で指定します。

- パッケージの `open` メソッドに渡される引数として
- NVRAM 変数 `network-boot-arguments` 内に指定する引数として

`network-boot-arguments` 内に指定された引数は、パッケージの `open` メソッドに引数が 1 つも渡されなかった場合にのみ処理されます。

引数の値

protocol には、使用するアドレス検索プロトコルを指定します。`rarp`、`dhcp` のいずれかを指定できます。

このドキュメントで定められた新しい構文とスタイルを使って他の構成パラメータが指定されていた場合、*protocol* パラメータが指定されていなければ、手動の構成を意味します。

他の構成パラメータが 1 つも指定されていない場合や、他の構成パラメータが現在サポートされている位置に基づくパラメータ構文を使って指定されていた場合、*protocol* パラメータを指定しなければ、ネットワークブートサポートパッケージは、プラットフォーム固有のデフォルトのアドレス検索プロトコルを使用することになります。

手動構成を行う場合、IP アドレス、ブートファイル名、ブートファイルイメージの提供元サーバーのアドレスの各情報を、クライアントに提供する必要があります。また、ネットワークの構成によっては、`subnet-mask` と `router-ip` も指定する必要があります。

protocol 引数が指定されなかった場合、ネットワークブートサポートパッケージは、プラットフォーム固有のデフォルトのアドレス検索プロトコルを使用します。

`tftp-server` は、TFTP を使用する場合、ダウンロードするファイルを提供する TFTP サーバーの IP アドレス (標準 IPv4 のドット区切り 10 進表記) です。

DHCP を使用する場合、その値は、DHCP 応答に指定された TFTP サーバーの値よりも優先されます。

サーバーが引数として指定された場合と DHCP 応答内に指定された場合、TFTP RRQ はサーバーにユニキャストされます。それ以外の場合、TFTP RRQ はブロードキャストされます。

file には、TFTP を使用する場合は TFTP サーバーから読み込むファイルを指定し、HTTP を使用する場合は URL を指定します。ファイル名が URL である場合、すなわち、ファイル名が `http:` で始まる場合 (大文字小文字の区別あり)、HTTP が使用されます。

RARP と TFTP を使用する場合、このドキュメントで前述したように、デフォルトのファイル名はクライアントの IP アドレスの ASCII 16 進表記になります。

DHCP を使用する場合、この引数は、DHCP 応答に指定されたブートファイル名よりも優先されます。

DHCP と TFTP を使用する場合、デフォルトのファイル名は、root ノードの `name` プロパティーから自動生成されます。その際、コンマ (,) はピリオド (.) に置き換えられます。

ファイル名をコマンド行から指定する場合、ファイル名にスラッシュ (/) を含めることはできません。

URL の形式については、RFC 2396 で説明されています。HTTP サーバーは、IP アドレス (標準 IPv4 のドット区切り 10 進表記) として指定する必要があります。ポート番号 (省略可能) は、10 進数として指定します。ポートを指定しなかった場合、ポート 80 (10 進) が使用されます。

指定された URL は「安全にエンコードされている」必要があります。というのも、パッケージは、指定された URL にエスケープエンコーディングを適用しないからです。コンマを含む URL は、引用符付きの文字列として指定する必要があります。それ以外の URL は、必ずしも引用符で囲む必要はありません。

`host-ip` には、クライアントすなわちブート対象システムの、IP アドレス (標準 IPv4 のドット区切り 10 進表記) を指定します。アドレス検索プロトコルとして RARP を使用している場合にこの引数を指定すると、RARP を使用する必要がなくなります。

DHCP を使用している場合に `host-ip` 引数を指定すると、クライアントは、RFC 2131の「Externally Configured Network Address」で規定されている手順に従うようになります。

`router-ip` には、直接接続されたネットワーク上にあるルーターの IP アドレス (標準 IPv4のドット区切り 10 進表記) を指定します。このルーターは、ネットワーク通信の最初の接続先として使用されます。この引数を指定した場合、そこで指定されたルーターは、DHCP 応答に指定された推奨ルーターよりも優先されます。

`subnet-mask` (標準 IPv4 のドット区切り 10 進表記で指定) は、クライアントが存在するネットワークのサブネットマスクです。サブネットマスクが (この引数または DHCP 応答を通じて) 指定されなかった場合、ブート対象クライアントに割り当てられたアドレスに対するネットワーククラス (Class A、B、C のいずれか) に適するデフォルトマスクが使用されます。

`client-id` には、クライアントの一意に決まる識別子を指定します。DHCP クライアント識別子は、この値に基づいて生成されます。クライアント識別子の指定方法には、次の 2 つがあります。

- 識別子の ASCII 16 進表現
- 引用符付きの文字列

したがって、「`client-id="openboot"`」と「`client-id=6f70656e626f6f74`」はどちらも、DHCP クライアント識別子 6F70656E626F6F74 を表わします。

コマンド行から指定される識別子には、スラッシュ (/) やスペースを含めることはできません。

DHCP クライアント識別子の最大長は、32 バイト (ASCII 16 進形式を使用する場合は 32 バイトを表す 64 文字) です。後者の形式を使用する場合、識別子を構成する文字数は偶数でなければなりません。有効な文字は、0 〜 9、a 〜 f、A 〜 F です。

クライアントが正しく認識されるためには、クライアントが接続されているサブネット上で、クライアント識別子が一意に決まる必要があります。この要件を満足する識別子を選択するのは、システム管理者の役割です。

コマンド行で指定されたクライアント識別子は、任意の DHCP メカニズムを使って指定された識別子よりも優先されます。

`hostname` (文字列として指定) には、DHCP トランザクション内で使用されるホスト名を指定します。この名前は、ローカルドメイン名で修飾してもしなくてもかまいません。ホスト名の最大長は、255 文字です。

注 – `hostname` パラメータは、クライアントが、希望するホスト名を DHCP サーバーに提供することが必要なサービス環境で使用できます。クライアントが、希望するホスト名を DHCP サーバーに提供すると、DHCP サーバーは、そのホスト名とクライアントに割り当てられた IP アドレスを、DNS を使って登録します。

`http-proxy` は、次のようなホストの標準記法を使って指定します。

host [`":""` *port*]

...ここで、*host* には IP アドレス (標準 IPv4 のドット区切り 10 進表記) を指定し、*port* (省略可能) には 10 進数を指定します。ポートを指定しなかった場合、ポート 8080 (10 進) が使用されます。

`tftp-retries` は、TFTP 処理が失敗したとみなされるまでの、最大リトライ回数 (10進で指定) です。デフォルトのリトライ回数は、無限回です。

`dhcp-retries`は、DHCP 処理が失敗したとみなされるまでの、最大リトライ回数 (10 進で指定) です。デフォルトのリトライ回数は、無限回です。

x86 でのブートストラップ手続き

x86 システム上では、ブートストラップ手続きは、カーネルのロードと、カーネルの初期化という概念的に異なる 2 つのフェーズからなります。カーネルのロードは、システムボード上の BIOS ROM および周辺装置上の BIOS ROM 拡張機能を使用して、GRUB (GRand Unified Bootloader) 上で実装されます。BIOS は、ハードディスク、DVD、または CD の先頭の物理セクターから開始して GRUB をロードします。ネットワークアダプター上の ROM でサポートされている場合は、BIOS はネットワークサーバーから `pxegrub` バイナリをダウンロードすることもできます。GRUB は配置されると、メニューにあるコマンドを実行して `UNIX` カーネルと、カーネルモジュールやデータを含む、事前に構成されたブートアーカイブをロードします。

GRUB によってブートデバイスとして識別されたデバイスに ZFS ストレージプールが含まれている場合は、GRUB メニューを作成するために使用される `menu.lst` ファイルが、プールのデータセット階層のルートにあるデータセット内で検索されます。 これは、プール自体と同じ名前のデータセットです。プール内にはこのようなデータセットが常に正確に 1 つ存在するため、このデータセットは、`menu.lst`

ファイルなどのプール全体にわたるデータに適しています。システムがブートされたあと、このデータセットは、ルートファイルシステム内の */poolname* にマウントされます。

プール内に複数のブート可能データセット (つまり、ルートファイルシステム) が存在することがあります。デフォルトでは、`menu.lst` ファイル内のファイル名のエントリが解決されるファイルシステムは、そのプールの `bootfs` プロパティーによって識別されるファイルシステムです (`zpool(1M)` を参照)。ただし、`menu.lst` のエントリに `bootfs` コマンドを含めて、プールの代替データセットを指定することもできます。このように、`menu.lst` ファイルには、プール内の複数のルートファイルシステムに対応するエントリが含まれている場合があります。

カーネルの初期化は、GRUB がブートアーカイブの読み込みを完了し、`UNIX` バイナリに制御を渡した時点で開始されます。この時点で、GRUB は非アクティブとなり、ブートデバイスとの入出力はこれ以上発生しません。UNIX オペレーティングシステムは、ルートファイルシステムを初期化し、ブートアーカイブの必要なモジュール内にリンクして、実際のルートデバイスにマウントします。この時点で、カーネルは、ストレージに対する入出力を再開し、さらなるファイルシステムをマウントします (`vfstab(4)` を参照)。そして、多様なオペレーティングシステムのサービスを開始します (`smf(5)` を参照)。

自動リブートの有効化 (x86)

Solaris オペレーティングシステムでは、古いブートアーカイブなどの状況から回復するために、現在のブートデバイスからシステムを自動的にリブートできる `smf(5)` プロパティーがサポートされています。

サービス `svc:/system/boot-config:default` には、ブール型のプロパティー `auto-reboot-safe` (デフォルトで `false` の設定) が含まれます。これを `true` に設定すると、システムの BIOS およびデフォルトの GRUB メニューエントリの両方が、現在のブートデバイスからブートされるよう設定するための通信が行われます。このプロパティーの値は、`svccfg(1M)` および `svcadm(1M)` を使用して変更できます。たとえば、自動リブートを有効にするように `auto-reboot-safe` を設定するには、次のようにコマンドを入力します。

```
example# svccfg -s svc:/system/boot-config:default \
      setprop config/auto-reboot-safe = true
```

ほとんどのシステムは、現在のブートデバイスから自動的にリブートするように構成されています。ただし、場合によって、不明なオペレーティングシステムへの自動リブートにより、好ましくない結果が生じることがあります。このような場合に `auto-reboot-safe` プロパティーを使用すると、目的の動作を指定できます。

フェイルセーフモード

ブートアーカイブに組み込まれたルートファイルシステムのイメージからブートし、ルートを実際のルートデバイスに再マウントするためには、ブートアーカイブとルートファイルシステムの内容に一貫性が必要です。 そうでない場合は、マシンの正しい動作や整合性を保証できません。

「一貫性がある」という用語は、ルートファイルシステム内のすべてのファイルとモジュールがブートアーカイブ内にも存在し、かつ同一の内容であることを示します。このブート方式では、最初にブートアーカイブを読み取って第 1 段階のルートイメージとしてマウントすることが必要なため、実際のルートファイルシステムと一致させるには、ロード不可能なすべてのカーネルモジュールと、ブートアーカイブの内容に基づいた初期化が必要です。このような一貫性がない場合、システムの実行に際して、リブート前にルートデバイスには適用されたがルートアーカイブではまだ更新されていないカーネルモジュールやパラメータが使用される可能性があります。この不一致のために、システムが不安定になったり、データが損失したりすることがあります。

ルートファイルシステムがマウントされたあと、メモリー内ファイルシステムを放棄する前に、Solaris はこの 2 つのファイルシステムに対する一貫性検証を実行します。不一致が検出された場合、Solaris は通常のブートシーケンスを中断し、フェイルセーフモードにフォールバックします。この状態を修正するには、管理者が次の 2 つの手順のうちの 1 つを実行する必要があります。推奨される手順は、フェイルセーフアーカイブにリブートし、ブートアーカイブを再構築するこことです。これにより、既知のカーネルがブートされ、アーカイブの再構築処理に対して機能することが保証されます。あるいは、この不一致によって正しいシステム動作が損なわれることがない場合、管理者は、一貫性のないブートアーカイブのサービス状態をクリアーし、システムの動作を続行することを選択できます。`svcadm(1M)` を参照してください。

ブートアーカイブサービスがクリアーされ、システムの動作が続行された場合 (上の 2 番目の選択)、システムは、ルートファイルシステムに関して期限が切れたロード不可能なカーネルドライバまたはその他のモジュールを使用して実行される可能性があります。そのため、正しいシステム動作が損なわれることがあります。

ブートアーカイブの一貫性を保証するために、`reboot(1M)` や `shutdown(1M)` によって開始される通常のシステムシャットダウン処理では、`umountall(1M)` マイルストーンの完了時にブートアーカイブへの更新をチェックし、適用します。

`umountall` サービスが完了したあとにブートアーカイブに含める必要のあるカーネルファイル、ドライバ、モジュール、またはドライバ構成ファイルになんらかの更新が発生すると、次回のブート中にブートアーカイブの一貫性検査は失敗します。この事態を回避するために、常にマシンをクリーンな状態で停止することをお勧めします。

`umountall` サービスの完了後にカーネルへの更新が必要な場合、管理者は、次の呼び出しによってアーカイブを再構築することを選択できます。

```
# bootadm update-archive
```

フェイルセーフブートアーカイブ

フェイルセーフアーカイブを使用すると、保守またはトラブルシューティングのために、マシンを任意の時点でブートできます。フェイルセーフブートアーカイブは、ミニルートアーカイブをソースとしてマシンにインストールされます。フェイ

ルセーフアーカイブをブートすると、マシンは、メモリー内ファイルシステムをルートデバイスとして使用してブートされます。

SPARC

SPARC フェイルセーフアーカイブは次のとおりです。

```
/platform/‘uname -i‘/failsafe
```

...また、次のようにブートできます。

```
ok boot [device-specifier] -F failsafe
```

ユーザーが特定の ZFS ブート可能データセットからのフェイルセーフアーカイブのブートを希望する場合は、これを次のように実行できます。

```
ok boot [device-specifier] -Z dataset -F failsafe
```

x86

x86 フェイルセーフアーカイブは次のとおりです。

```
/boot/x86.miniroot-safe
```

...また、GRUB メニューから `Solaris` フェイルセーフの項目を選択することによってブートできます。

オプション

SPARC

次の SPARC オプションがサポートされています。

`-a`
: boot プログラムはこのフラグを問い合わせと解釈し、スタンドアロンプログラムの名前を要求します。`'-a'` フラグはその後、スタンドアロンプログラムに渡されます。

`-D` *default-file*
: *default-file* を明示的に指定します。一部のシステムでは、なにも指定されていない場合、`boot` は動的デフォルトファイルを選択します。このオプションを使用すると、*default-file* を明示的に設定できます。`kmdb(1)` をブートする場合に便利です。これは、`kmdb` がデフォルトで、`boot` プログラムによってエクスポートされたデフォルトファイルを読み込むためです。

`-F` *object*
: 名前付きオブジェクトを使用してブートします。このオブジェクトは、ELF 実行可能ファイル、またはブートブロックを含むブート可能なオブジェクトのどちらかである必要があります。主に、フェイルセーフまたは `wanboot` ブートアーカイブのブートに使用します。

`-L`
: ZFS プール内のブート可能データセットを一覧表示します。リスト内のいずれかのブート可能データセットを選択できます。そのあと、そのデータセットを

ブートするための詳細な手順が表示されます。その手順に従って、選択したデータセットをブートします。 このオプションは、ブートデバイスに ZFS ストレージプールが含まれている場合にのみサポートされます。

`-V`
詳細なデバッグ情報を表示します。

boot-flags
boot プログラムはすべての *boot-flags* を `file` に渡します。これらの引数は、`boot` では解釈されません。デフォルトのスタンドアロンプログラムで使用できるオプションについては `kernel(1M)` および `kmdb(1)` のマニュアルページを参照してください。

client-program-args
`boot` プログラムはすべての *client-program-args* を *file* に渡します。これらの引数は、`boot` では解釈されません。

file
ブートするスタンドアロンプログラムの名前。`boot` コマンド行または *boot-file* NVRAM 変数でファイル名を明示的に指定しないと、`boot` は適切なデフォルトファイル名を選択します。

OBP names
OpenBoot PROM 指定を行います。たとえば、Desktop SPARC ベースのシステム上で `/sbus/esp@0,800000/sd@3,0:a` を指定した場合は、スロット 0 の esp ホストアダプタで、SCSI バス上の lun0 、ターゲット 3 の SCSI ディスク (sd) を意味します。

`-Z` *dataset*
指定した ZFS データセット内のルートファイルシステムからブートします。

x86 次の x86 オプションがサポートされています。

`-B` *prop=val...*
カーネルに渡される 1 つ以上の属性値ペア。複数の設定値ペアはコンマで区切られる必要があります。このオプションの使用方法は、次のコマンドと同様です。`eeprom` *prop=val*設定情報とその有効な値については、 `eeprom(1M)` を参照してください。

このメニューエントリに対応するルートファイルシステムが ZFS データセットである場合は、メニューエントリに次のオプションを追加する必要があります。

```
-B $ZFS-BOOTFS
```

boot-args
boot プログラムはすべての *boot-args* を `file` に渡します。これらの引数は、`boot` では解釈されません。カーネルで使用できるオプションについては、`kernel(1M)` および `kmdb(1)` を参照してください。

特に指定しないかぎり、x86 システムでは /platform/i86pc/kernel/amd64/unix がブートされます。

x86 でのブートシーケンスの詳細

PC 互換マシンの電源を投入すると、BIOS ROM のシステムファームウェアが電源投入時自己診断テスト (POST) を実行し、周辺ボード上の ROM の BIOS 拡張機能を実行し、ソフトウェア割り込み INT 19h のブートストラップをブートします。INT 19h のハンドラは通常、標準の PC 互換ブートを実行します。その場合、1 番目のハードディスクから先頭物理セクターの読み取りを試みます。次にプロセッサは、メモリー内でこのセクターイメージの先頭バイトにジャンプします。

x86 での一次ブート

ディスクメディアの先頭セクターにはマスターブートブロック (GRUB ステージ 1) があります。ステージ 1 は GRUB ステージ 2 をロードします。この時点で GRUB は準備が完全に整います。GRUB は、メニューファイル (/boot/grub/menu.lst) を読み込み、実行します。同様のシーケンスが DVD あるいは、CD ブートでも発生します。しかし、マスターブートブロックの位置と内容は El Torito 仕様で規定されています。El Torito 準拠のブートでも、strap.com が読み込まれ、それが次に boot.bin が読み込まれます。

ハードディスクの最初のセクターには、マスターブートブロックがあり、そのブロックにはマスターブートプログラムと FDISK テーブルが含まれており、PC プログラムがそのテーブルを管理しています。マスターブートは、 FDISK テーブルの中のアクティブパーティションを探し、その先頭セクター (GRUB ステージ 1) をロードして、メモリー内の先頭バイトにジャンプします。ハードディスクからの標準の PC 互換ブート処理はこれで完了します。GRUB ステージ 1 がマスターブートブロック上にインストールされると (installgrub(1M) の -m を参照)、ステージ 2 が、アクティブパーティションかどうかにはかかわらず Solaris FDISK パーティションから直接ロードされます。

Solaris ソフトウェア用 x86 FDISK のパーティションは、1 シリンダのブートスライスから始まり、これは、GRUB のステージ 1 を先頭セクターに含んでいます。そして、第 2 および第 3 のセクターは、標準 Solaris ディスクラベルおよびボリューム構成テーブル (VTOC) を含んでいます。そして、第 50 とその後続セクターは、GRUB のステージ 2 を含んでいます。第 4 から 第 49 までのセクターには、Solaris の古いバージョンのブートブロックが含まれている可能性があります。これによって、同じ FDISK の中に、複数の Solaris リリースを同時に存在させることが可能になります。Solaris ソフトウェア用 FDISK パーティションがアクティブパーティションの場合、マスターブートプログラム (mboot) は、先頭セクターにあるパーティションブートプログラムをメモリーの中に読み込み、そこへジャンプします。次に GRUB ステージ 2 プログラムをメモリーに読み込み、そこへジャンプします。GRUB メニューが表示されると、ユーザーは異なるパーティション、異なるディスク、あるいは 可能であればネットワークからオペレーティングシステムをブートできるよう選択できます。

Intel の Preboot eXecution Environment (PXE) 規格が、ネットワークブートのためにサポートされています。PXE を使用してネットワークからのブートを行う場合には、システムあるいはネットワークアダプタの BIOS が DHCP を使用してブート

サーバ上のネットワークブートストラッププログラム (pxegrub) を見つけ、Trivial File Transfer Protocol (TFTP) を使用してそのプログラムを読み込みます。BIOS が、メモリ上の先頭バイトにジャンプすることで、pxegrub を実行します。pxegrub は、メニューファイルをダウンロードし、ユーザーにその項目を表示します。

x86 カーネルの開始

カーネルの開始処理は、カーネルのロード処理に依存しません。カーネルが開始している間、コンソール入出力は、console プロパティーで指定されたデバイスで行われます。

UFS からブートする場合、ルートデバイスは bootpath プロパティーで指定され、ルートファイルシステムの種類は fstype プロパティーで指定されます。これらのプロパティーは、Solaris インストールおよびアップグレード処理で/boot/solaris/bootenv.rc の中に設定されるべきです。そして、前述のように (eeprom(1M) を参照) -B で上書きされます。

ZFS からブートする場合、ルートデバイスは、GRUB メニューエントリ内の kernel または module のどちらかの行にある -B $ZFS-BOOTFS パラメータが指定するブートパラメータで指定されます。この値は、-B オプションで指定されるすべてのパラメータと同様に、GRUB によってカーネルに渡されます。

コンソールプロパティーが存在しない場合、コンソールの入出力は、デフォルトで screen と keyboard になります。ルートデバイスはデフォルトで ramdisk になり、ファイルシステムはデフォルトで ufs になります。

使用例

SPARC

例 1　対話モードでシングルユーザーとしてデフォルトのカーネルをブートする

対話モードでシングルユーザーとしてデフォルトのカーネルをブートするには、ok プロンプトで次のいずれかを入力します。

```
boot -as
```

```
boot disk3 -as
```

例 2　WANブートに対応した PROM によるネットワークブート

さまざまな boot コマンド行呼び出し間の微妙な違いを示すために、ここでは network-boot-arguments が設定されており、以下のコマンドのように net に対してデバイス別名が設定されているものとします。

次のコマンドでは、デバイス別名内のデバイス引数が、デバイスドライバによって処理されます。ネットワークブートサポートパッケージは、network-boot-arguments 内の引数を処理します。

```
boot net
```

次のコマンドでは、デバイス引数なし、とみなされます。ネットワークブートサポートパッケージは、network-boot-arguments 内の引数を処理します。

例 2　WANブートに対応したPROMによるネットワークブート　　(続き)

```
boot net:
```

次のコマンドでは、デバイス引数なし、とみなされます。rarp は唯一のネットワークブートサポートパッケージ引数です。network-boot-arguments は無視されます。

```
boot net:rarp
```

次のコマンドでは、指定されたデバイス引数が処理されます。ネットワークブートサポートパッケージは、network-boot-arguments 内の引数を処理します。

```
boot net:speed=100,duplex=full
```

例 3　以前の PROM における wanboot の使用

次のコマンドでは、DVD または CD から wanboot バイナリが読み込まれます。読み込まれた wanboot は、DHCP の実行時にコマンドインタプリタをブートし、鍵やその他の必要な構成情報をユーザーが入力できるようにします。

```
boot cdrom -F wanboot -o dhcp,prompt
```

x86

例 4　対話モードでシングルユーザーとしてデフォルトのカーネルをブートする

シングルユーザー状態の対話モードでデフォルトのカーネルをブートするには、コマンド行で GRUB を次のように編集します。

```
kernel /platform/i86pc/kernel/amd64/unix -as
```

ファイル

/etc/inittab
: initdefault 状態が指定されているテーブル

/usr/sbin/init
: initdefault 状態にシステムを移行するプログラム

SPARC のみ

/platform/*platform-name*/kernel/sparcv9/unix
: システムをブートするデフォルトのプログラム。

x86 のみ

/boot
: ブートに関連するファイルが置かれているディレクトリ。

/rpool/boot/grub/menu.lst
: GRUB によって表示されるブート可能なオペレーティングシステムのメニュー。/rpool は共通する規約です。パス名は、インストーラの機能に応じて構成可能です。

/platform/i86pc/kernel/amd64/unix
: システムをブートするデフォルトのプログラム。

関連項目

kmdb(1), uname(1), bootadm(1M), eeprom(1M), init(1M), installboot(1M), kernel(1M), monitor(1M), shutdown(1M), svcadm(1M), svccfg(1M), umountall(1M), zpool(1M), uadmin(2), bootparams(4), inittab(4), vfstab(4), wanboot.conf(4), attributes(5), filesystem(5), smf(5)

RFC 903, A Reverse Address Resolution Protocol, http://www.ietf.org/rfc/rfc903.txt

RFC 2131, Dynamic Host Configuration Protocol, http://www.ietf.org/rfc/rfc2131.txt

RFC 2132, DHCP Options and BOOTP Vendor Extensions, http://www.ietf.org/rfc/rfc2132.txt

RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, http://www.ietf.org/rfc/rfc2396.txt

『Oracle Solaris の管理: 一般的なタスク』

『Sun ハードウェアマニュアル』

『OpenBoot Command Reference Manual』

警告

boot ユーティリティは、ブート可能プログラムとして使用できるファイルかどうか判別できません。ブート不可能なファイルをブートするように要求された場合、boot ユーティリティはそのファイルをロードし、そのファイルに制御を渡します。その場合の結果は予測できません。

注意事項

platform-name を調べるには、uname(1) の -i オプションを使用します。*hardware-class-name* を調べるには、uname(1) の -m オプションを使用します。

現在のリリースの Solaris オペレーティングシステムは、UltraSPARC-I CPU を搭載するマシンをサポートしません。

名前

bootadm – GRUB 対応オペレーティングシステムのブート管理

形式

```
/usr/sbin/bootadm update-archive [-vn] [-R altroot [-p platform]]

/usr/sbin/bootadm list-archive [-vn] [-R altroot [-p platform]]

 x86 only

/usr/sbin/bootadm set-menu [-R altroot [-p platform]] key=value

/usr/sbin/bootadm list-menu [-R altroot [-p platform]]
```

機能説明

bootadm コマンドは、ブートアーカイブのほか、x86 ブート環境、GRUB (GRand Unified Bootloader) メニューを管理します。update–archive オプションを使えば、ブートアーカイブの更新を、障害予防策として、あるいは復旧手順の一環として、実施できます。set-menu サブコマンドを使えば、GRUB メニュー内の自動ブートタイムアウトやデフォルトブートエントリを切り替えることができます。

list-menu サブコマンドは、GRUB メニューの場所と現在のエントリ内容を表示します。ZFS での GRUB メニューの場所は通常は /rpool/boot/grub/menu.lst ですが、使用されたインストール方法によっては、アクティブな GRUB メニューがほかの場所に格納されている可能性もあります。list-menu サブコマンドを使ってアクティブな GRUB メニューの場所を特定してください。たとえば、Live Upgrade を使ってインストールされたシステムの場合、現在のブート環境内に GRUB メニューが格納されていない可能性があります。list-menu オプションの典型的な出力については、「使用例」の節を参照してください。

SPARC システムなどの OpenBoot PROM (OBP) ベースのマシンは GRUB を使用しないため、bootadm で管理できるブートメニューがないことに注意してください。

あるプラットフォーム上で呼び出される bootadm を使用して、プラットフォームの種類が異なるディスクレスクライアントを管理できるように、bootadm コマンドは、管理対象のイメージでサポートされるオプションを動的に判断します。

サブコマンド

bootadm コマンドのサブコマンドを、次に示します。

update-archive
: 現在のブートアーカイブを必要に応じて更新します。SPARC と x86 の両方のプラットフォームに適用されます。

list-archive
: ブートアーカイブに含めるファイルとディレクトリを一覧表示します。SPARC と x86 の両方のプラットフォームに適用されます。

set-menu
: GRUB メニューを維持管理します。現在の GRUB メニューは、boot/grub/menu.lst にあります。ただし、これはルートからの相対パスです。これは変更される可能性があるため、この場所には依存しないでください。x86 プラットフォームにのみ適用されます。

`list-menu`
アクティブな GRUB メニューの場所と現在の GRUB メニューエントリを一覧表示します。これには、自動ブートタイムアウト、デフォルトエントリ番号、および各エントリのタイトルが含まれます。x86 プラットフォームにのみ適用されます。

オプション

`bootadm` コマンドには、次のオプションがあります。

`-v`
`update-archive` 処理の場合、無効なファイルが標準エラー出力に表示されます。

`-n`
`update-archive` 処理の場合、アーカイブ内容のチェックは行われますが、その更新は行われません。

`-p` *platform*
クライアントのプラットフォームまたはマシンのハードウェアクラス。プラットフォームの種類は `-R` とともにのみ指定でき、一般にはクライアントのプラットフォームクラスがサーバーとは異なる場合のディスクレスクライアントの管理にのみ有効です。プラットフォームには、`i86pc`、`sun4u`、`sun4v` のいずれかを指定する必要があります。

`-R` *altroot*
別のルートパスに処理が適用されます。

注 – いかなる非大域ゾーンのルートファイルシステムも `-R` で参照しないでください。この操作によって、大域ゾーンのファイルシステムを損傷したり、大域ゾーンのセキュリティを損ねたり、非大域ゾーンのファイルシステムを損傷する可能性があります。`zones(5)` を参照してください。

key=value
可能な値は次のとおりです。

`default=`*entrynum*
GRUB メニュー内の項目番号 (0、1、2 など)。タイムアウト時にブートするオペレーティングシステムを指定します。

`timeout=`*seconds*
デフォルト項目番号で指定されたオペレーティングシステムがブートされるまでの秒数。値が –1 の場合、自動ブートは無効になります。

使用例

例 1 現在のブートアーカイブの更新

次のコマンドは、現在のブートアーカイブを更新します。

```
# bootadm update-archive
```

例 2 別のルート上にあるブートアーカイブの更新

次のコマンドは、別のルート上にあるブートアーカイブを更新します。

```
# bootadm update-archive -R /a
```

例 3 インストール済み OS インスタンスの一覧表示

次のコマンドは、GRUB メニューに含まれている、インストール済みのオペレーティングシステムインスタンスを一覧表示します。

```
# bootadm list-menu

default=0
timeout=10
(0) Solaris10
(1) Solaris10 Failsafe
(2) Linux
```

例 4 デフォルトブートエントリの切り替え

次のコマンドは、1 つ前の例で表示されたメニューを参照しています。ユーザーは、Linux (項目 2) を選択しています。

```
# bootadm set-menu default=2
```

例 5 GRUB メニューのエントリと場所の一覧表示

次のコマンドは、GRUB メニューのエントリと場所を一覧表示しています。

```
# bootadm list-menu
The location for the active GRUB menu is: /stubboot/boot/grub/menu.lst
default 0
timeout 10
0 Solaris10
1 Solaris10 failsafe
2 Linux
```

例 6 GRUB メニューの場所の表示

次のコマンドは、GRUB メニューの場所を表示しています。

```
# bootadm list-menu
The location for the active GRUB menu is: /dev/dsk/c0t1d0s0 (not mounted)
The filesystem type of the menu device is <ufs>
default 2
timeout 10
0 c0t1d0s3
1 c0t1d0s3 failsafe
2 Solaris10
3 Solaris10 failsafe
```

例 6 GRUB メニューの場所の表示 (続き)

この例では、アクティブな GRUB メニューが、マウントされていないデバイス上に格納されています。この GRUB メニューにアクセスするには、このデバイスをマウントし、<マウントポイント>/boot/grub/menu.lst にある GRUB メニューにアクセスします。

終了ステータス

次の終了値が返されます。

0
コマンドが正常に完了しました。

1
エラーが発生したため、コマンドが終了しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

boot(1M), installgrub(1M), attributes(5)

次の GRUB ホームページを参照してください。

http://www.gnu.org/

名前

bootconfchk – verify the integrity of a network boot configuration file

形式

`/usr/sbin/bootconfchk [bootconf-file]`

機能説明

The `bootconfchk` command checks that the file specified is a valid network boot configuration file as described in `wanboot.conf(4)`.

Any discrepancies are reported on standard error.

終了ステータス

`0` Successful completion.

`1` An error occurred.

`2` Usage error.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Committed

関連項目

`wanboot.conf(4)`, `attributes(5)`

名前

busstat – report bus-related performance statistics

形式

```
busstat -e device-inst | -h | -l

busstat [-a] [-n]
     [-w  device-inst [,pic0=event,picn=event ]]...
     [-r device-inst]... [interval  [count]]
```

機能説明

`busstat` provides access to the bus-related performance counters in the system. These performance counters allow for the measurement of statistics like hardware clock cycles, bus statistics including DMA and cache coherency transactions on a multiprocessor system. Each bus device that supports these counters can be programmed to count a number of events from a specified list. Each device supports one or more Performance Instrumentation Counters (PIC) that are capable of counting events independently of each other.

Separate events can be selected for each PIC on each instance of these devices. `busstat` summarizes the counts over the last interval seconds, repeating forever. If a count is given, the statistics are repeated count times.

Only root users can program these counters. Non-root users have the option of reading the counters that have been programmed by a root user.

The default value for the *interval* argument is 1 second, and the default *count* is unlimited.

The devices that export these counters are highly platform-dependent and the data may be difficult to interpret without an in-depth understanding of the operation of the components that are being measured and of the system they reside in.

オプション

The following options are supported:

`-a`	Display absolute counter values. The default is `delta` values.
`-e` *device-inst*	Display the list of events that the specified device supports for each pic. Specify *device-inst* as device (`name`) followed by an optional instance number. If an instance number is specified, the events for that instance are displayed. If no instance number is specified, the events for the first instance of the specified device are displayed.
`-h`	Print a usage message.
`-l`	List the devices in the system which support performance counters.
`-n`	Do not display a title in the output. The default is to display titles.

-r *device-inst*	Read and display all pic values for the specified device Specify *device-inst* as *device* (name) followed by *instance number*, if specifying an instance number of a device whose counters are to be read and displayed. If all instances of this device are to be read, use *device* (name) without an instance number. All pic values will be sampled when using the -r option.
-w *device-inst* [,pic0=*event*] [,pic*n*=*event*]	Program (write) the specified devices to count the specified events. Write access to the counters is restricted to root users only. Non-root users can use -r option. Specify *device-inst* as *device* (name) followed by an optional *instance number*. If specifying an instance number of a device to program these events on. If all instances of this device are to be programmed the same, then use *device* without an instance number. Specify an event to be counted for a specified pic by providing a comma separated list of pic*n*=*event* values. The -e option displays all valid event names for each device. Any devices that are programmed will be sampled every interval seconds and repeated count times. It is recommended that the interval specified is small enough to ensure that counter wraparound will be detected. The rate at which counters wraparound varies from device to device. If a user is programming events using the -w option and busstat detects that another user has changed the events that are being counted, the tool will terminate as the programmed devices are now being controlled by another user. Only one user can be programming a device instance at any one time. Extra devices can be sampled using the -r option. Using multiple instances of the -w option on the same command line, with the same *device-inst* specifying a different list of events for the pics will give the effect of multiplexing for that device. busstat will

switch between the list of events for that device every interval seconds. Event can be a string representing the event name, or even a number representing the bit pattern to be programmed into the Performance Control Register (PCR). This assumes explicit knowledge of the meaning of the control register bits for a device. The number can be specified in hexadecimal, decimal, or octal, using the usual conventions of `strtol`(3C).

終了ステータス

The following exit values are returned:

0 Successful completion.

1 An error occurred.

2 Another user is writing to the same devices.

使用例

SPARC Only

例 1 Programming and monitoring the Address Controller counters

In this example, `ac0` refers to the Address Controller instance 0. The counters are programmed to count Memory Bank stalls on an Ultra Enterprise system at 10 second intervals with the values displayed in absolute form instead of deltas.

```
# busstat -a -w ac0,pic0=mem_bank0_stall,pic1=mem_bank1_stall 10
time  dev   event0            pic0   event1             pic1
10    ac0   mem_bank0_stall   1234   mem_bank1_stall    5678
20    ac0   mem_bank0_stall   5678   mem_bank1_stall   12345
30    ac0   mem_bank0_stall  12345   mem_bank1_stall   56789
...
```

For a complete list of the supported events for a device, use the `-e` option.

例 2 Programming and monitoring the counters on all instances of the Address Controller

In this example, `ac` refers to all `ac` instances. This example programs all instances of the Address Controller counters to `count_clock` cycles and `mem_bank0_rds` at 2 second intervals, 100 times, displaying the values as deltas.

```
# busstat -w ac,pic0=clock_cycles,pic1=mem_bank0_rds 2 100
time  dev     event0          pic0            event1          pic1
2     ac0     clock_cycles    167242902       mem_bank0_rds   3144
2     ac1     clock_cycles    167254476       mem_bank0_rds   1392
4     ac0     clock_cycles    168025190       mem_bank0_rds   40302
4     ac1     clock_cycles    168024056       mem_bank0_rds   40580
...
```

例 3 Monitoring the events being counted

This example monitors the events that are being counted on the sbus1 device, 100 times at 1 second intervals. It suggests that a root user has changed the events that `sbus1` was counting to be `dvma_tlb_misses` and interrupts instead of `pio_cycles`.

```
% busstat -r sbus0 1 100

time    dev     event0              pic0       event1          pic1
1       sbus1   pio_cycles          2321       pio_cycles      2321
2       sbus1   pio_cycles          48         pio_cycles      48
3       sbus1   pio_cycles          49         pio_cycles      49
4       sbus1   pio_cycles          2281       pio_cycles      2281
5       sbus1   dvma_tlb_misses     0          interrupts      0
6       sbus1   dvma_tlb_misses     6          interrupts      2
7       sbus1   dvma_tlb_misses     8          interrupts      11
...
```

例 4 Event Multiplexing

This example programs `ac0` to alternate between counting (clock cycles, `mem_bank0_rds`) and (`addr_pkts`, `data_pkts`) at 2 second intervals while also monitoring what `ac1` is counting :

It shows the expected output of the above `busstat` command. Another root user on the machine has changed the events that this user had programmed and `busstat` has detected this and terminates the command with a message.

```
# busstat -w ac0,pic0=clock_cycles,pic1=mem_bank0_rds \
           -w ac0,pic0=addr_pkts,pic1=data_pkts \
           -r ac1 2

time    dev     event0          pic0            event1          pic1
2       ac0     addr_pkts       12866           data_pkts       17015
2       ac1     rio_pkts        385             rio_pkts        385
4       ac0     clock_cycles    168018914       mem_bank0_rds   2865
4       ac1     rio_pkts        506             rio_pkts        506
6       ac0     addr_pkts       144236          data_pkts       149223
6       ac1     rio_pkts        522             rio_pkts        522
8       ac0     clock_cycles    168021245       mem_bank0_rds   2564
8       ac1     rio_pkts        387             rio_pkts        387
10      ac0     addr_pkts       144292          data_pkts       159645
10      ac1     rio_pkts        506             rio_pkts        506
12      ac0     clock_cycles    168020364       mem_bank0_rds   2665
12      ac1     rio_pkts        522             rio_pkts        522
busstat: events changed (possibly by another busstat).
#
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

iostat(1M), mpstat(1M), vmstat(1M), strtol(3C), attributes(5)

名前

captoinfo – convert a termcap description into a terminfo description

形式

`captoinfo [-1] [-v]... [-V] [-w` *width*`]` *filename*`...`

機能説明

`captoinfo` looks in *filename* for `termcap` descriptions. For each one found, an equivalent `terminfo` description is written to standard output, along with any comments found. A description which is expressed as relative to another description (as specified in the `termcap` `tc = field`) is reduced to the minimum superset before being displayed.

If no *filename* is given, then the environment variable `TERMCAP` is used for the filename or entry. If `TERMCAP` is a full pathname to a file, only the terminal whose name is specified in the environment variable `TERM` is extracted from that file. If the environment variable `TERMCAP` is not set, then the file `/usr/share/lib/termcap` is read.

オプション

`–1`	Display the fields one to a line. Otherwise, the fields are printed several to a line, with a maximum width of 60 characters.
`-v`	Display tracing information on the standard error as the program runs. Specifying additional `-v` options displays more detailed information.
`-V`	Display the version of the program in use on the standard error and then exit.
`-w` *width*	Change the output to *width* characters.

ファイル

`/usr/share/lib/terminfo/?/*`	compiled terminal description database
`/usr/share/lib/termcap`	

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`infocmp(1M)`, `curses(3CURSES)`, `terminfo(4)`, `attributes(5)`

注意事項

`captoinfo` should be used to convert `termcap` entries to `terminfo` entries because the `termcap` database may not be supplied in future releases.

名前

catman – リファレンスマニュアル用の書式付きファイルの作成

形式

```
/usr/bin/catman [-c] [-n] [-p] [-t] [-w] [-M directory]
      [-T macro-package] [sections]

/usr/bin/catman [-M directory] -w
```

機能説明

catman ユーティリティーは、nroff(1) または sgml(5) の入力ファイルからオンラインマニュアルの事前整形されたバージョンを作成します。事前整形されたマニュアルページのディレクトリは、自己完結的にかつ書式なしエントリから独立して作成されるので、一群のマシンの間でそれらのマニュアルページを容易に配布できます。

また、-w オプションを使用すると、catman は MANPATH または -M オプションで指定されたディレクトリにインデックスファイルを作成します。MANPATH または -M オプションが指定されていない場合、-n を指定しなければ、catman はデフォルトで /usr/share/man/ および /usr/gnu/share/man/ ディレクトリにインデックスファイルを作成します。指定されたディレクトリまたはデフォルトのディレクトリが読み取り専用の場合、catman は失敗し、そのディレクトリへの書き込みが許可されていないことを示すエラーメッセージを標準エラー出力に表示します。

各マニュアルページが検査され、事前整形されたバージョンが見つからないもの、または最新ではないものについては再作成されます。なんらかの変更が行われた場合、catman はインデックスファイルを再作成します。

マニュアルページが *shadow* (シャドウ) ページである場合、つまりその内容が書かれた別のマニュアルページをソースとしている場合、該当する事前整形されたマニュアルページへのシンボリックリンクが cat*x* または fmt*x* ディレクトリに作成されます。

書式なし nroff のソースファイルにあるシャドウファイルは、最初の行に .so man*x*/yyy.*x* という形式の行があることで識別できます。

SGML のソースファイルにあるシャドウファイルは、文字列 SHADOW_PAGE があることで識別できます。シャドウファイルで宣言されたファイルの実体は、ソースとなるファイルを示しています。

オプション

次のオプションがサポートされています。

-c
: SGML のソースファイルから、書式なし nroff のソースファイルを適切な man サブディレクトリに作成します。このオプションは SGML ファイルと同じ名前で man ディレクトリにある既存のファイルをすべて上書きします。

-n
: インデックスファイルを作成 (再作成) しません。-n オプションを指定した場合、インデックスファイルは作成されず、apropos(1) および whatis(1) コマンドの実行速度が、それ以外の場合よりも低下する可能性があります。

-p
予行演習オプション。つまり、実行するのではなく何が行われるのかを表示します。

-t
cat サブディレクトリへの nroff を行う代わりに、該当する fmt サブディレクトリ内に troff 処理後のエントリを作成します。

-w
apropos(1)、whatis(1)、および man(1) の -f、-k、および -K オプションで使用されるインデックスファイルを、MANPATH 環境変数または -M オプションで指定されるディレクトリに作成します。MANPATH または -M オプションを指定しない場合、インデックスファイルはデフォルトで /usr/share/man/ および /usr/gnu/share/man/ ディレクトリに作成されます。マニュアルの再整形は実行されません。

-M *directory*
引数に指定したディレクトリ (デフォルトは /usr/share/man) 中にあるマニュアルページを更新します。コンマは、マニュアルセクションの番号を示すために使用されているので、-M オプションの引数として指定するディレクトリ名には ',' (コンマ) が含まれないようにしてください。man(1) を参照してください。

-T *macro-package*
標準のマニュアルページマクロの代わりに、 *macro-package* 引数で指定したマクロパッケージを使用します (デフォルトは man(5))。

オペランド

次のオペランドがサポートされています。

sections
先頭文字が '–' でないパラメータは、catman によって処理されるマニュアルセクションと見なします。各セクションはスペースで区切ります。このオペランドが指定されると、指定するマニュアルセクションだけが処理されます。たとえば、次のように指定します。

```
catman 1 2 3
```

前述のコマンドは、セクション 1、2、3 だけを更新します。セクションが指定されない場合、環境変数 MANPATH に指定された man ディレクトリにあるすべてのセクションが処理されます。

環境変数

TROFF
-t オプションが指定された場合に用いるフォーマッタの名前。この環境変数が設定されていない場合は、troff(1) が使用されます。

MANPATH
catman と man(1) で処理されるディレクトリの、コロンで区切られたリスト。各ディレクトリは、コンマで区切られたセクションのリストの後に続いて指定できます。この環境変数に値を設定すると、デフォルトのディレクトリ検索パスである /usr/share/man と、デフォルトのセクション検索パスである man.cf を無効にします。-M および -s フラグはそれぞれこれらの値を無効にします。

使用例

例1 インデックスファイルの作成

次のコマンドは、インデックスファイルを /usr/local/share/man ディレクトリに作成します。

```
# catman -M /usr/local/share/man -w
```

ファイル

/usr/share/man
マニュアルページのデフォルトのディレクトリ

/usr/share/man/man*/*.*
nroff 入力ファイル (未処理)

/usr/share/man/sman*/*.*
SGML 入力ファイル (未処理)

/usr/share/man/cat*/*.*
事前整形された nroff 処理後のマニュアルページ

/usr/share/man/fmt*/*.*
事前整形された troff 処理後のマニュアルページ

/usr/share/lib/tmac/an
デフォルトのマクロパッケージ

/usr/share/man/man_index/*
/usr/share/man/man_index/man.idx
/usr/share/man/man_index/man.dic
/usr/share/man/man_index/man.frq
/usr/share/man/man_index/man.pos
-K クエリー用のインデックスファイル

属性

次の属性については、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	text/doctools
CSI	有効
インタフェースの安定性	確実

関連項目

apropos(1), man(1), nroff(1), rm(1), troff(1), whatis(1), attributes(5), man(5), sgml(5)

診断

man?/xxx.? (.so'ed from man?/yyy.?): No such file or directory
メッセージの冒頭に示されたファイルは、カッコ内のファイルが参照しようとしたが、存在しませんでした。

target of .so in man?/xxx.? must be relative to /usr/man
catman では、ディレクトリ /usr/man からの相対パスで示したファイル名のみを参照できます。

opendir:man?: No such file or directory
: catman が通常検索するディレクトリのうちの 1 つが見つかりません (軽度の警告メッセージ)。

.: No such file or directory
: catman によって、 中身が空のディレクトリが見つかりました (軽度の警告メッセージ)。

警告

以前に catman を実行して cat* ディレクトリがすでにインストールされている場合に、オペレーティングシステムをアップグレードした場合は、catman を実行する前に cat* ディレクトリ構造全体を削除してください (rm(1) を参照)。

すべての man* ディレクトリがそろっていない場合は、インデックスファイルを再作成するために catman を再度実行しないでください。catman は、man* ディレクトリに基づいてインデックスファイルを作成します。

注意事項

windex データベースはインデックスファイルに置き換えられました。windex を使用した場合と異なり、インデックスファイルの生成では、インデックスを作成できる対象について特定の制限や前提条件は課せられません。

名前

cfgadm – 構成の管理

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options]
     -c function ap_id...

/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options]
     -x hardware_function ap_id...

/usr/sbin/cfgadm [-v] [-a] [-s listing_options]
     [-o hardware_options] [-l [ap_id | ap_type]]

/usr/sbin/cfgadm [-v] [-o hardware_options] -t ap_id...

/usr/sbin/cfgadm [-v] [-o hardware_options] -h
     [ap_id | ap_type]
```

機能説明

`cfgadm` コマンドを使用して、動的な再構成が可能なハードウェアリソースに対して構成の管理を行うことができます。これらの操作には、ステータス (state) の表示 (`-l`)、検査の開始 (`-t`)、構成ステータスの変更の開始 (`-c`)、ハードウェア固有の機能の実行 (`-x`)、および構成管理のヘルプ情報の表示 (`-h`)、が含まれます。構成管理は、接続点 (*attachment point*) で実行されます。接続点は、Solaris の動作中にハードウェアリソースの動的再構成を行うことにシステムソフトウェアが対応している場所です。

構成の管理では、マシン上に実際にあるハードウェアリソースと、構成済みで Solaris が認識できるハードウェアリソースが区別されます。構成管理機能の特性はハードウェアに依存し、ハードウェア固有のライブラリを呼び出すことで実行されます。

構成管理は、接続点で実行されます。接続点に設置されているハードウェアリソースには、システムの稼働中に物理的な交換ができるものとできないものがありますが、構成管理インタフェースによって、動的に再構成することはできます。

接続点は、接続点の向こう側に位置するハードウェアリソースとは別の2 つの固有の要素を定義します。接続点の 2 つの要素は、受容体(*receptacle*) と占有装置 (*occupant*) です。ハードウェアリソースの物理的な取り付け、取り外しは接続点で行われ、その結果、受容体に占有装置が追加されたり削除されたりします。構成管理は、接続点での構成管理機能だけでなく、この物理的な着脱操作にも対応しています。

接続点には、状態 (state) と条件 (condition) の情報が関連付けられています。構成管理インタフェースを使用して、接続点の状態の変化を制御することができます。受容体は、 `empty`、`disconnected`、`connected` の 3 つの状態のいずれかになります。また、占有装置は、 `configured` と `unconfigured` のいずれかの状態になります。

受容体は、接続点に占有装置がない場合に、必ず受容体の通常の状態である `empty` になります。この状態には、稼働中のシステムの一部を一時的に停止することができる、ハードウェア固有の機能が関係します。受容体が占有装置をシステムの通常の使用から切り離すことができる場合に、その受容体は `disconnected` 状態になるこ

ともできます。この状態は、占有装置のハードウェアリソースをシステムが完全に利用できるようにする前にそのハードウェアに対する検査を実行する場合や、占有装置の物理的な取り外しや再構成のための準備の 1 つの段階として、主に使用されます。disconnected 状態の受容体は、ハードウェアの許容範囲内で占有装置をシステムから分離しますが、検査や設定が必要な場合は使用を許可する場合があります。受容体は、占有装置に含まれるハードウェアリソースの通常の使用が許可されている場合に、必ず connected 状態になります。connected 状態は、占有装置を含み、かつ構成管理操作が実行されていない受容体の通常の状態です。

unconfigured 状態の占有装置に含まれるハードウェアリソースは、Solaris の通常のデータ構造では表現されないため、Solaris はそのハードウェアを使用できません。未構成の占有装置に対して実行できる操作は、構成管理操作に限られています。configured 状態の占有装置に含まれるハードウェアリソースは、Solaris の通常のデータ構造で表現されるため、Solaris は、一部またはすべてのハードウェアリソースを使用することができます。占有装置は、必ず configured 状態か unconfigured 状態になります。

接続点は、 unknown、ok、failing、failed、unusable の 5 つの条件のいずれかになります。接続点は、電源投入検査と不揮発性記録保存の結果によって、システムをどの条件にも置く可能性があります。

configured 状態の占有装置を持つ接続点は、 unknown、ok、failing、failed の条件のいずれかになります。failing または failed 条件にない接続点は、ハードウェア固有の回復可能なエラーがしきい値を超えると、操作中に failing 状態になる場合があります。また、failed 条件にない接続点は、回復不可能なエラーによって、操作中に failed 条件に変わる場合があります。

unconfigured 状態にある占有装置を持つ接続点は今までに挙げた条件のどれになる可能性もあります。unconfigured 状態にある占有装置を持つ接続点の条件は、マシン固有の時間しきい値が経過した後に、ok から unknown になる場合があります。検査機能を開始した場合は、検査の結果によって接続点の条件が ok、failing、failed のいずれかに変わります。検査機能を持たない接続点は、接続点を unknown 条件のままにする場合があります。検査が中断された場合は、接続点の条件は、以前の条件、unknown、failed に設定することができます。unknown、ok、failing、failed のいずれかの条件にある接続点には、再検査を行うことができます。

接続点は、さまざまな理由によって unusable 条件になります。理由としては、受容体に対する不適切な電力投入や冷却、占有装置が認識できない、対応していない、不適切に構成されている、などが挙げられます。unusable 条件にある接続点は、システムで使用することができません。通常、この条件は、接続点に対して物理的な対処がなされない限り変わりません。

また、接続点は、状態の変更処理が進行中である場合や、条件が再評価されている場合に、それを示す使用状態情報を保持します。

接続点は、システムデバイス階層構造の中での接続点のタイプと位置に関連するハードウェア固有の識別子 (*ap_ids*) に対応しています。*ap_id* は単一の接続点を特定するために、一意になっている必要があります。*ap_id* の仕様には、 物理タイプと論理タイプの 2種類が用意されています。物理 *ap_id* には、完全なパス名を指定します。論理 *ap_id* には、簡略表記法を使用し、ユーザーにとってより簡単な方法で接続点を指定します。

たとえば、システムのバックプレーンスロット番号 7 にある接続点の物理 *ap_id* は `/devices/central/fhc/sysctrl:slot7` となり、論理 *ap_id* は `system:slot7` になります。また、システムの第 2 PCI 入出力バス上にある 3 番目の受容体の論理 *ap_id* は `pci2:plug3` になります。

接続点も動的に作成されます。動的接続点には、そのシステムに設定されている基本接続点を基にして名前が付けられます。動的接続点の `ap_ids` は、2 つのコロン (`::`)、基本コンポーネント、および動的コンポーネントで構成されます。基本コンポーネントは、基本接続点 `ap_id` です。動的コンポーネントは、ハードウェア固有で、対応するハードウェア固有のライブラリによって生成されます。

たとえば、SCSI HBA を表現し、物理 `ap_id` が`/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi` で、論理 `ap_id` が c0 である基本接続点を想定します。この SCSI HBA に接続されているディスクは、論理 `ap_id` が `c0::dsk/c0t0d0` である動的接続点によって表現されます。ここで、`c0` は基本コンポーネントで、`dsk/c0t0d0` はハードウェア固有の動的コンポーネントです。同様に、この動的接続点の物理 `ap_id` は `/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi::dsk/c0t0d0` になります。

ap_type は `ap_id` の一部で、それ自身だけでは一意にならず、単一の接続点を特定することができません。*ap_type* は、論理 `ap_id` の一部を含み、コロン (`:`) 区切り記号を含まない部分文字列です。たとえば、`pci` の *ap_type* は、論理 *ap_id* が `pci` で始まる接続点をすべて出力します。

ap_types は、できるだけ使用しないでください。`-s` オプションの新しい選択サブオプションを使用すれば、より汎用的な方法で柔軟に接続点を選択することができます。詳しくは「オプション」の節を参照してください。

`cfgadm` コマンドは、ハードウェア固有ライブラリに含まれるハードウェア固有の機能と主に対話するため、cfgadm コマンドの動作はハードウェアに依存します。

それぞれの構成管理操作では、サービスの中断が必要になる場合があります。要求された機能を完了するために、対話式で操作中のユーザーにとって目に見えるサービスの中断が必要になる場合は、機能の開始前に標準エラー出力に確認メッセージが表示され、標準入力による確認を促します。すべての質問に対する yes を意味する `-y` オプション、または no を意味する `-n` オプションを指定することによって、確認を省略することもできます。検査レベルなどのハードウェア固有のオプションは `-o` オプションを使用して、サブオプションとして指定することができます。

システム構成の状態を変更する操作は、システムログデーモンである syslogd(1M) によって監視されます。

このコマンドの引数は、getopt(3C) および getsubopt(3C) の構文規約に従います。

オプション

次のオプションがサポートされています。

-a

-l オプションによって、動的接続点のリストも出力されるように指定します。

-c*function*

ap_id で指定された接続点の状態を *function* で指定された状態に変更します。

function には、insert、remove、disconnect、connect、configure、unconfig のいずれかを指定することができます。これらの関数は、ハードウェア固有のライブラリルーチンを呼び出して接続点の状態を変更します。これらの関数の定義を以下に示します。

insert

占有装置を手動で追加する操作を実行したり、物理的な追加を実行するハードウェア機能を起動します。insert には、システムの一部を一時的に停止するハードウェア固有の副作用が伴う場合があります。このような場合、ハードウェア固有のライブラリは、対応する警告メッセージを生成し、ユーザーに対して、そのハードウェア固有の問題と手順を提供します。ハードウェアに起因するさまざまなエラーによってこの関数が失敗し、受容体の条件が unusable になる場合があります。

remove

占有装置を手動で削除する操作を実行したり、物理的な削除を実行するハードウェア機能を起動します。remove には、システムの一部を一時的に停止するハードウェア固有の副作用が伴う場合があります。このような場合、ハードウェア固有のライブラリは、対応する警告メッセージを生成し、ユーザーに対して、そのハードウェア固有の問題と手順を提供します。ハードウェアに起因するさまざまなエラーによってこの関数が失敗し、受容体の条件が unusable になる場合があります。

`disconnect` ハードウェア固有の操作を実行して、受容体を disconnected 状態にします。disconnected 状態にすることによって、占有装置に対して、受容体を介した通常の方法による操作が行えなくなります。

`connect` ハードウェア固有の操作を実行して、受容体を `connected` 状態にします。受容体が connect 状態にすることによって、占有装置に対して、受容体を介した通常の方法による操作が行えるようになります。

`configure` ハードウェア固有の操作を実行して、占有装置のハードウェアリソースを Solaris が使用できるようにします。構成された占有装置はシステム構成の一部になり、`psradm(1M)`、`mount(1M)`、`ifconfig(1M)` などの Solarisデバイス操作メンテナンスコマンドによる操作の対象となります。

`unconfigure` ハードウェア固有の操作を実行して、占有装置のハードウェアリソースをシステムから論理的に削除します。この関数を使用するには、占有装置が現在構成されていて、占有装置のハードウェアが Solaris によって使用されていない必要があります。

状態変更関数は、接続点の条件や、その他のハードウェア固有の問題によって失敗する場合があります。リソースを追加するための状態変更関数 (`insert`、`connect`、`configure`) は、接続点が `ok` または `unknown` 条件にある場合に、ハードウェア固有のライブラリに渡されます。接続点がそれ以外の条件にある場合は、強制オプション (-f) を使用した場合に限って、資源を追加するための関数がハードウェア固有のライブラリに渡されます。システムからハードウェアリソースを削除するための関数 (`remove`、`disconnect`、`unconfigure`) によるハードウェア固有のライブラリの呼び出しは、接続点の条件によって妨げられることはありません。接続点が `unknown` 条件にある場合に、関数は、ハードウェア固有のライブラリによって拒否される場合があります。

接続点の条件は、状態変更関数によって変更されないこともありますが、状態変更操作中のエラーによって接続点の条件が変わる場合があります。条件の書き換えと状態の強制的な変更は、強制オプション (-f) を指定した場合だけ実行することができます。これらの処理は、強制オプションを指定しないと失敗します。強制オプションは、ハードウェア固有の安全性検査および整合性検査によって無効になる場合があります。

-f

指定された処理を強制的に実行します。これは主に、ハードウェア固有の安全機能を無効にするために使用します。状態の変更操作を強制することによって、ハードウェア固有の安全検査によって ok や unknown 状態にない占有装置のハードウェアリソースを使用することができる場合があります。

-h [*ap_id* | *ap_type* . . .]

ヘルプメッセージテキストを出力します。*ap_id* または *ap_type* を指定すると、この引数によって指定された接続点に関するハードウェア固有のライブラリのヘルプルーチンが呼び出されます。

-l [*ap_id* | *ap_type* . . .]

指定された接続点の状態や条件を一覧表示します。接続点を抽出するには、-s オプションと select サブオプションを使用します。いずれかの処理オプションを使用せずに cfgadm コマンドを起動するのは、引数を使用せずに -l を使用するのと同等です。表示画面の書式は -v および -s オプションによって制御されます。-a オプションが指定されているときは、接続点が動的に展開されます。

-n

対話型の確認を行わず、応答が *no* であるとみなします。-n と -y のいずれも指定しないと、標準エラー出力と標準入力によって対話型の確認が行われます。これらの標準的なチャネルのいずれも端末 (isatty(3C) によって判定されている) に対応していない場合は、-n オプションが想定されます。

-o*hardware_options*

コマンドの主オプションに対してハードウェア固有のオプションを指定します。hardware_options の文字列の書式と内容は完全にハードウェア固有のものです。オプション文字列の *hardware_options* は getsubopt(3C) の構文規約に従います。

-s*listing_options*

一覧表示 (-l) コマンドに対して一覧表示オプションを指定します。*listing_options* は getsubopt(3C) の構文規約に従います。サブオプションを使用して、接続点の選択条件 (select=*select_string*)、適切な照合タイプ

(match=*match_type*)、一覧表示する順序 (sort=*field_spec*)、表示するデータ (cols=*field_spec* と cols2=*field_spec*)、列の区切り記号 (delim=*string*)、列の見出し行の抑制 (noheadings) を指定します。

select サブオプションを指定すると、指定された条件と一致する接続点だけが一覧表示されます。select サブオプションの構文は次のとおりです。

```
cfgadm -s select=attr1(value1):attr2(value2)...
```

attr は、ap_id、class、type のいずれかです。ap_id は論理 *ap_id* フィールド、class は接続点のクラス、type はタイプフィールドです。*value1*、*value2* などは、照合する値です。照合タイプを指定するには、次のように match サブオプションを使用します。

```
cfgadm -s match=match_type,select=attr1(value1)...
```

match_type は、exact または partial から選択します。デフォルト値は exact です。

select サブオプションの引数は、シェルから保護するために引用符で囲みます。

field_spec には、1 つの *data-field*、または次のようにコロン (:) で区切った複数の *data-field* を指定します。*data-field*: *data-field*: data-field。*data-field* は、ap_id、physid、r_state、o_state、condition、type、busy、statu のいずれかです。ap_id フィールドの出力は接続点の論理名で、physid フィールドは物理名です。r_state フィールドは、empty、disconnected、connected のいずれかになります。o_state フィールドは、configured と unconfigured のいずれかになります。busy フィールドは、接続点が使用中の場合に y になり、使用中でない場合に n となります。type フィールドと info フィールドはハードウェア固有のフィールドです。status_time フィールドには、r_state、o_state、または接続点の条件が最後に変更された時刻が表示されます。status_time_p フィールドは、構文解析が可能な status_time フィールドです。接続点にクラスが関連付けられている場合は、class フィールドにクラス名が表示されます。接続点にクラスが関連付けられていない場合は、class フィールドに none が表示されます。

field_spec 内のフィールドの順序は重要です。 sort サブオプションでは、最初に与えられたフィールドが主ソート

キーになります。cols および cols2 サブオプションでは、指定した順序でフィールドが出力されます。*data-field* に対するソートの順序は、sort サブオプションに対する *field_sec* 内の *data-field* 名の前にマイナス (–) を付けることによって逆になります。sort のデフォルトの値は ap_id です。cols および cols2 のデフォルトの値は -v オプションが指定されているかどうかによって異なります。 v が指定されていない場合は、cols は ap_id:r_state:o_state:condition になり、cols2 は設定されません。-v が指定されている場合は、cols は ap_id:r_state:o_state:condition:info になり、cols2 は status_time:type:activity:physid: になります。delim のデフォルトの値は、単一のスペース文字です。delim の値には、任意の長さの文字列を指定することもできます。区切り記号にはコンマ (,) を含めることはできません (getsubopt(3C) を参照)。これらの一覧表示オプションは、構文解析が可能な出力を生成するために使用することができます。「注意事項」 を参照してください。

-t

1 つまたは複数の接続点の検査を実行します。この検査機能は、接続点の条件を再評価するために使用します。*hardware_options* の中で検査レベル指示子を指定しないと、重度の障害を検出する最も早い検査方法が使用されます。

より包括的な検査は個々のハードウェアに依存するため、*hardware_options* を使用して選択します。

検査の結果は、指定された占有装置の条件を ok (障害が発見されなかった場合)、failing (回復可能な障害が発見された場合)、failed (回復不可能な障害が発見された場合) のいずれかに更新するために使用されます。

検査が中断された場合は、接続点の条件は、以前の値に戻すか、unknown (障害が発見されなかった場合)、failing (回復可能な障害が発見された場合)、failed (回復不可能な障害が発見された場合) のいずれかに設定されます。接続点は、エラーがなく、検査が正常に完了した場合のみ ok に設定されます。

-v

詳細表示モードで実行します。-c、-t、-x オプションを指定した場合に、各試行操作の結果を表示するメッセージを出力します。-h オプションを指定した場合

は、詳細なヘルプ情報が表示されます。-l オプションを指定した場合は、各接続点に関するすべての情報を出力します。

-x*hardware_function*　ハードウェア固有の機能を実行します。受容体や占有装置の状態は、専用ハードウェア固有の機能を使用して変更します。接続点の状態は、ハードウェア固有の機能の動作中に検出されたエラーの結果として変化する場合があります。*hardware_function* 文字列の書式と内容は完全にハードウェア固有のものです。オプション文字列の *hardware_function* は getsubopt(3C) の構文規約に従います。

-y　対話型の確認を行わず、応答が yes であるとみなします。

使用法

このコマンドを使用するために必要な特権は、ハードウェアに依存します。一般的に、デフォルトのシステム設定では、一覧表示オプション以外のすべての機能はスーパーユーザーのみが使用することができます。

使用例

例 1　デバイスツリー内の接続点の一覧表示

以下の例は、動的接続点以外のすべての接続点を一覧表示します。

```
example# cfgadm

  Ap_Id          Type         Receptacle      Occupant        Cond
system:slot0     cpu/mem      connected       configured      ok
system:slot1     sbus-upa     connected       configured      ok
system:slot2     cpu/mem      connected       configured      ok
system:slot3     unknown      connected       unconfigured    unknown
system:slot4     dual-sbus    connected       configured      failing
system:slot5     cpu/mem      connected       configured      ok
system:slot6     unknown      disconnected    unconfigured    unusable
system:slot7     unknown      empty           unconfigured    ok
c0               scsi-bus     connected       configured      unknown
c1               scsi-bus     connected       configured      unknown
```

例 2　構成することができるすべてのハードウェアの情報の一覧表示

以下の例は、現在構成することができるすべてのハードウェアの情報を一覧表示します。動的接続点で表現されるハードウェアも表示されます。

```
example# cfgadm -al

  Ap_Id            Type         Receptacle      Occupant        Cond
system:slot0       cpu/mem      connected       configured      ok
system:slot1       sbus-upa     connected       configured      ok
system:slot2       cpu/mem      connected       configured      ok
system:slot3       unknown      connected       unconfigured    unknown
```

例 2 構成することができるすべてのハードウェアの情報の一覧表示 *(続き)*

```
system:slot4       dual-sbus    connected       configured      failing
system:slot5       cpu/mem      connected       configured      ok
system:slot6       unknown      disconnected    unconfigured    unusable
system:slot7       unknown      empty           unconfigured    ok
c0                 scsi-bus     connected       configured      unknown
c0::dsk/c0t14d0    disk         connected       configured      unknown
c0::dsk/c0t11d0    disk         connected       configured      unknown
c0::dsk/c0t8d0     disk         connected       configured      unknown
c0::rmt/0          tape         connected       configured      unknown
c1                 scsi-bus     connected       configured      unknown
```

例 3 接続点の属性に基づいて選択された情報の一覧表示

次の例は、接続点のうち、scsi で始まるクラス、c で始まる ap_id、および scsi で始まる type フィールドを持つものをすべて一覧表示します。-s オプションの引数は、シェルから保護するために引用符で囲みます。

```
example# cfgadm -s "match=partial,select=class(scsi):ap_id(c):type(scsi)"

Ap_Id          Type          Receptacle      Occupant           Cond
 c0           scsi-bus      connected       configured         unknown
 c1           scsi-bus      connected       configured         unknown
```

例 4 現在構成することができるハードウェアの情報の詳細表示

以下の例は、*ap-type* system の現在構成することができるハードウェアの情報を詳細表示モードで示します。

```
example# cfgadm -v -l system
Ap_Id                      Receptacle Occupant   Condition Information
When         Type      Busy     Phys_Id
system:slot1               connected  configured ok
Apr  4 23:50 sbus-upa  n        /devices/central/fhc/sysctrl:slot1
system:slot3               connected  configured ok        non-detachable
Apr 17 11:20 cpu/mem   n        /devices/central/fhc/sysctrl:slot3
system:slot5               connected  configured ok
Apr  4 23:50 cpu/mem   n        /devices/central/fhc/sysctrl:slot5
system:slot7               connected  configured ok
Apr  4 23:50 dual-sbus n        /devices/central/fhc/sysctrl:slot7
```

When 列は status_time フィールドの内容を表示します。

例 5 ハードウェア固有の拡張検査を使用した 2 つの占有装置の検査

次の例は、ハードウェア固有の拡張検査を使用して 2 つの占有装置を検査します。

例 5 ハードウェア固有の拡張検査を使用した 2 つの占有装置の検査 (続き)

```
example# cfgadm -v -o extended -t system:slot3 system:slot5
Testing attachment point system:slot3 ...  ok
Testing attachment point system:slot5 ...  ok
```

例 6 強制オプションを使用した占有装置の構成

以下の例は、強制オプションを使用して、failing 状態の占有装置をシステムに構成します。

```
example# cfgadm -f -c configure system:slot3
```

例 7 システムからの占有装置の構成解除

以下の例は、システムから占有装置を構成解除します。

```
example# cfgadm -c unconfigure system:slot4
```

例 8 接続点の占有装置の構成

以下の例は、占有装置を構成します。

```
example# cfgadm -c configure c0::dsk/c0t0d0
```

環境変数

cfgadm の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。 LC_TIME、LC_MESSAGES、NLSPATH、および TZ。

LC_MESSAGES	cfgadm が見出し行とエラーメッセージを表示する方法を特定します。出力されるデータは、この環境変数の影響を受けません。
LC_TIME	cfgadm がステータスの変更時間 (status_time) を表示する方法 (人による判読が可能な形式) を特定します。
TZ	ステータスの変更時間を変換する際に使用されるタイムゾーンを特定します。これは人による判読が可能な (status_time) と構文解析が可能な (status_time_p) 形式の両方に当てはまります。

終了ステータス

次の終了値が返されます。

0	正常終了。
1	エラーが発生しました。
2	指定された宛先が構成管理に対応していません。
3	使用方法のエラー。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

cfgadm_fp(1M), cfgadm_ib(1M), cfgadm_pci(1M), cfgadm_sata(1M), cfgadm_sbd(1M), cfgadm_scsi(1M), cfgadm_usb(1M), ifconfig(1M), mount(1M), prtdiag(1M), psradm(1M), syslogd(1M), config_admin(3CFGADM), getopt(3C), getsubopt(3C), isatty(3C), attributes(5), environ(5)

診断

診断メッセージは標準エラー出力に表示されます。オプションや使用方法のエラー以外に、次の診断メッセージがこのユーティリティによって表示されます。

`cfgadm: Configuration administration not supported on`*ap_id*

`cfgadm: No library found for` *ap_id*

`cfgadm:` *ap_id*`is ambiguous`

`cfgadm:` *operation*`: Insufficient privileges`

`cfgadm: Attachment point is busy, try again`

`cfgadm: No attachment points with specified attributes found`

`cfgadm: System is busy, try again`

`cfgadm:` *operation*`: Operation requires a service interruption`

`cfgadm:` *operation*`: Data error:` *error_text*

`cfgadm:` *operation*`: Hardware specific failure:` *error_text*

エラーメッセージの詳細については config_admin(3CFGADM) を参照してください。

注意事項

ハードウェアリソースは、ハードウェア固有の方法で未構成プールに入ります。これは、 システムの初期設定や構成解除操作の結果としてなど、さまざまな状況で発生します。unconfigured 状態にある占有装置は、システムによる特定の介入が発生するまで、システムが使用することはできません。このような干渉は、オペレーターが開始したコマンドや、自動構成メカニズムによって発生します。

cfgadm コマンドの一覧表示オプションは、シェルスクリプトの中などで、別のコマンドに対する構文解析可能な入力として使用することができます。構文解析可能な出力の場合は、必要なフィールドを選択するときに、-s オプションを使用する必要があります。-s オプションは、列の見出しを抑制するときにも使用できます。次のフィールドは、常に構文解析が可能な出力を生成します。
ap_id、physid、r_state、o_state、condition、busy、status_time_p、class、および type。解析可能な出力には、フィールドの値にスペース文字が含まれることはありません。

以下はシェルスクリプトの一部分で、タイプが CPU の正常な未構成の占有装置で最初のものを検出します。

```
found=
cfgadm -l -s "noheadings,cols=ap_id:r_state:condition:type" | \
while read ap_id r_state cond type
do
     if [ "$r_state" = unconfigured -a "$cond" = ok -a "$type" = CPU ]
     then
          if [ -z "$found" ]
          then
               found=$ap_id
          fi
     fi
done
if [ -n "$found" ]
then
         echo "Found CPU $found"
fi
```

構文解析が可能な時間フィールド (`status_time_p`) の形式は、*YYYYMMDDhhmmss* で、文字列の比較を行うのに都合がよい書式で、年、月、日、時間、分、秒を表示します。

システム構成管理で使用できるものの詳細については、ハードウェア固有のドキュメントを参照してください。

名前

cfgadm_ac – EXX00 メモリーシステムの管理

形式

```
/usr/sbin/cfgadm [-c configure] [-f]
     [-o disable-at-boot | enable-at-boot ] ac#:bank# ...

/usr/sbin/cfgadm [-c unconfigure]
     [-o disable-at-bootp | enable-at-boot ] ac#:bank# ...

/usr/sbin/cfgadm [-v]
     [-o quick | normal | extended, [max_errors=#] ] -t ac#:bank#...

/usr/sbin/cfgadm -x relocate-test ac#:bank# ...

/usr/sbin/cfgadm [-l] -o disable-at-boot | enable-at-boot ac#:bank# ...
```

機能説明

ac ハードウェア固有ライブラリ /usr/platform/sun4u/lib/cfgadm/cfgadm_ac.so.1 は、cfgadm_sysctrl(1M) を使用する CPUU/メモリーボードの Dynamic Reconfiguration (DR: 動的再構成) の一部として、E6X00、E5X00、E4X00、E3X00 システムのメモリーバンクの構成と構成解除に関連する機能を提供します。

メモリーバンクは、デバイスツリー上の接続点として表されます。CPU/メモリーボードそれぞれに 2 つの接続点が用意されます。 1 つがボード上の各バンク (バンク 0 とバンク 1) 用になります。 バンクが空いている場合は、受容体の状態は empty (空き) になります。バンクが使用されている場合は、受容体の状態は connected (接続) になります。 受容体の状態が disconnected (接続解除) になることはありません。接続されているメモリーバンクの占有状態は、構成することも構成解除することもできます。占有状態が構成されている場合は、メモリーは Solaris によって使用されています。構成解除されている場合は、使用されていません。

オプション

コマンドオプションの詳細については、cfgadm(1M) を参照してください。

次のオプションがサポートされています。

-c configure | unconfigure

占有状態を変更します。引数が configure の場合は、メモリーが初期化され、Solaris のメモリープールに追加します。引数が unconfigure の場合は、Solaris によって使用されているメモリーを切り離します。CPU/メモリーボードをシステムから取り外すには、2 つのメモリーバンクを構成解除する必要があります。

ボード上のメモリーが disabled-at-boot (info フィールド参照) に指定されていて、-f (強制) オプションやブート許可フラグ (-o enable-at-boot) がいずれも指定されていない場合は、cfgadm コマンドは、構成操作を許可しません。構成操作には、初期化する必要のあるメモリーのサイズに応じて、少し時間がかかります。

システムに十分な使用可能メモリーがない (VM viability エラー)、または構成解除されるべきバンクに切り離せないメモリーがある (non-relocatable pages エラー) 場合、cfgadm コマンドは、構成解除操作を許可しません。再配置不可のページの現状は、info フィールドの一覧にある permanent によって表されます。Solaris が使用しているメモリの取り外しを行うと、システム負荷や補助記憶装置のページングの大きさによって、かなり時間がかかります。構成解除操作は、いつでも中止することができ、シグナルによるコマンドの中断によって、メモリーは完全に構成されている状態に戻ります。もし、タイムアウトまでの期間内に削除できるメモリーがない場合は、構成解除操作は、自動的に取り消されます。デフォルトのタイムアウト期間は 60 秒ですが、-o timeout=# オプションで変更することもできます。この数値を 0 にすると、タイムアウトは不許可になります。

-f

強制オプション。このオプションは、非揮発性の変数 disabled-memory-list で、disabled at boot (ブート不可) に指定された、構成されているメモリーバンクのブロックを無効にしたい場合に使用してください。『特記事項: *Sun Enterprise 6x00*、*5x00*、*4x00*、*3x00* システム』のマニュアルも参照してください。

-l

リストオプション。このオプションについての説明は、cfgadm(1M) を参照してください。

type フィールドは、常に *memory* です。

info フィールドには、以下に示すような空のメモリーバンクについての情報が保持されます。

```
slot# empty
```

slot# は、CPU/メモリーボードが挿入されているシステムスロットを示しています。たとえば、これがスロット 11 である場合、cfgadm が関連付けられたボードを操作するために使用する接続点は、sysctrl0:slot11 になります。info フィールドには、次に示すような接続されているバンクについての情報が保持されます。

```
slot# sizeMb|sizeGb [(sizeMb|sizeGb used)] base 0x###
      [interleaved #-way] [disabled at boot] [permanent]
```

バンクのサイズは、Mb か Gb か適切な単位で与えられます。メモリーがまだ使い切られていない場合、使用サイズが表示されます。物理ベースアドレスは、16 進数で与えられます。メモリーバンクが他のバンクによってインタリーブされている場合は、インタリーブファクター(因子) が出力されます。ボード上のメモリーが、非揮発性の変数 `disabled-memory-list` を使ってブート不可になっている場合は、そのことが表示されます。バンクに切り離せないメモリーがあると、固定 (permanent) として出力されます。

-o disable-at-boot | enable-at-boot

このオプションは、非揮発性の disabled-memory-list 変数を変更することを許可します。これらのオプションは、コマンド要求がない場合は、`-c` オプションの発行や、明白または暗示的なリスティングオプション -l と結合して、使用することができます。使用不可メモリーの一覧にあるボード上に構成されているメモリーブロックを無効にするには、 `configure` コマンドを `-o enable-at-boot` オプションで使用してください。

-o extended | normal | quick

テストレベルを指定するには、`-t` オプションを使用してください。

テストレベル `normal` では、各メモリーセルに `0` と `1` のいずれも格納できることを確認して、さらにすべてのセルが個別にアドレス可能であるかどうか検査します。テストレベル `quick` では、すべてのメモリーに `0` と `1` を書き込むテストだけを行うため、アドレス線の障害を発見することはできません。`extended` テストでは、近接したセル同士の干渉の問題をテストするパターンを使用します。デフォルトのテストレベルは、`normal` です。`-t` オプションを参照してください。

-o max_errors=#

`-t` オプションと共に使用して、許可するエラーの最大数を指定します。このオプションを指定しなかった場合は、デフォルト値の `32` が適用されます。

-o timeout=#	構成解除コマンドと共に使用して、自動取り消しによるタイムアウトを設定します。デフォルト値は 60 で、単位は秒です。数値が 0 の場合は、タイムアウトしません。
-t	構成されていないメモリーのバンクをテストします。-o quick | normal | extended オプションでテストレベルを指定してください。 メモリーバンクテストが実行できた場合は、cfgadm コマンドはステータス 0 (成功) で終了します。テストの結果は、接続点のためという条件で利用可能です。
-v	詳細表示オプションです。-t オプションと組み合わせて使用することにより、テストの進行状況と結果を詳細に表示できます。
-x relocate-test	指定されたメモリーバンクで使用されているメモリーの全ページに対し、構成解除コマンドで行われるように、再配置を試みます。この操作の成功は、バンクが構成解除できるかどうかを保証しません。この操作の失敗は、構成解除できなかったことを意味します。このオプションの用途は、テストのみに限定されます。

オペランド

次のオペランドがサポートされています。

ac#:bank#

メモリーバンクの接続点は、アドレスコントローラー (*ac#*) ドライバ (ac) のインスタンスによって作成されます。ac ドライバの 1 つのインスタンスは、各システムボードによって作成されますが、CPU/メモリーボードに関連付けられたインスタンスだけは、バンク 0 とバンク 1 の 2 つの接続点を作成します。

このフォームは、cfgadm(1M) によって与えられた論理 ap_id 指定と一致します。これに対応する物理的な ap_id は、次項の【ファイル】に記載してあります。

ac ドライバインスタンスの番号付けは、対応するボードのスロット番号とは関係はありません。完全な物理接続点の識別子には、fhc@ に続く、スロット番号の 2 倍を 16 進数で表したスロット番号が入ります。

ファイル

/devices/fhc@*,f8800000/ac@0,1000000:bank?	接続点
/usr/platform/sun4u/lib/cfgadm/cfgadm_ac.so.1	ハードウェア固有ライブラリファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library/platform

関連項目

cfgadm(1M), cfgadm_sysctrl(1M), config_admin(3CFGADM), attributes(5)

『*Sun Enterprise 6x00*、*5x00*、*4x00*、*3x00* システム *Dynamic Reconfiguration* ユーザーマニュアル』

『特記事項: *Sun Enterprise 6x00*、*5x00*、*4x00*、*3x00* システム』

注意事項

EXX00 システムの CPU/メモリーボードの動的再構成について、更に詳しい情報が必要な場合は、『*Sun Enterprise 6x00*、*5x00*、*4x00*、*3x00* システム *Dynamic Reconfiguration* ユーザーマニュアル』を参照してください。

名前

cfgadm_cardbus – cfgadm の CardBus ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f ] [-y  | -n ] [-v]
     [-o hardware_options] -c function ap_id [ap_id]

/usr/sbin/cfgadm [-f ] [-y  | -n ] [-v]
     [-o hardware_options] -x hardware_function ap_id
     [ap_id]

/usr/sbin/cfgadm [-v] [-s listing_options]
     [-o hardware_options] [-l [ap_id | ap_type]]

/usr/sbin/cfgadm [-v] [-o harware_options] -t ap_id [ap_id]

/usr/sbin/cfgadm [-v] [-o hardware_function] -h
     [ap_id|  ap_type]
```

機能説明

Solaris の CardBus スロットはホットプラグに対応しています。この機能では、`cfgadm` コマンドによって PCI ハードウェア固有のライブラリ `/usr/lib/cfgadm/pci.so.1` がサポートされています (`cfgadm(1M)` を参照)。

CardBus、PCI、CompactPCI、および PCI Express 間のホットプラグ管理モデルでは同様に動作します。使用法については、`cfgadm_pci(1M)` を参照してください。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

`cfgadm(1M)`, `config_admin(3CFGADM)`, `libcfgadm(3LIB)`, `attributes(5)`

『Oracle Solaris の管理: 一般的なタスク』

名前

cfgadm_fp – cfgadm のドライバ固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-n | -y] [-v] [-o hardware_options]
     -c function ap_id [ap_id]

/usr/sbin/cfgadm [-v] [-a] [-s listing_options]
     [-o hardware_options] [-l [ap_id]]

/usr/sbin/cfgadm [-v] [-o hardware_options] -h [ap_id]
```

機能説明

fp ポートドライバプラグイン /usr/lib/cfgadm/fp.so.1 は、cfgadm(1M) によるファイバチャネルファブリックデバイスノードの管理機能を提供します。cfgadm は、接続点で実行されます。接続点とは、ハードウェアリソースを動的に再構成できるシステム上の場所です。接続点についての詳細は、cfgadm(1M) を参照してください。

ファイバチャネルファブリックデバイスノードの管理では、各 fp ポートノードは、デバイスツリー上で接続点によって表されます。また、各ファイバチャネルデバイスは動的接続点によって表されます。接続点には、ap_id を通じて名前が付けられます。論理 ap_id と物理 ap_id の 2 種類の ap_id が定義されます。物理 ap_id は、物理パス名に基づいています。論理 ap_id は、より短く、ユーザーにわかりやすい名前です。fp ポートノードの場合、論理 ap_id は、対応するディスクコントローラ番号です。たとえば、c0 は一般的な論理 ap_id です。

ファイバチャネルデバイスは、World Wide Name (WWN) で指定されます。ディスクデバイスがコントローラ c0 に接続されている場合、ap_id は次のようになります。

```
c0::50020f2300006077
```

ここで、50020f2300006077 は特定のファイバチャネルデバイスのポート WWN を識別します。

ファイバチャネルのプライベートループポート、ファブリックポート、またはパブリックループポート上の各デバイスが検索され、デフォルトで Solaris で使用可能になります。ファイバチャネルのファブリックポートまたはパブリックループポートに接続されたデバイスは、アプリケーションまたはエンドユーザー操作を開始することによって Solaris で使用可能にできます。この操作は、管理ユーザーインタフェースによるデバイスのホットプラグ解除と同様です。アプリケーションまたはユーザーは /usr/lib/cfgadm/fp.so.1 ライブラリを使用して、libcfgadm でこのタスクを実現するインタフェースを提供できます。

現在接続されているファブリックデバイスの一覧は、接続点の形式で生成されます。

システム上の接続点の単純な一覧には、fp ポートノード上の接続点は含まれますが、ファイバチャネルデバイス上の接続点は含まれません。次の例では、一覧表示オプション (-l) に -a フラグを使用してファイバチャネルデバイスを一覧表示しています。

```
# cfgadm -l
     Ap_Id                Type         Receptacle   Occupant     Condition
     c0                   fc-fabric    connected    configured   unknown
     c1                   fc-private   connected    configured   unknown
     c2                   fc-pt_to_pt  connected    configured   unknown
     c3                   fc           connected    unconfigured unknown
     sysctrl0:slot0       cpu/mem      connected    configured   ok
     sysctrl0:slot1       sbus-upa     connected    configured   ok
```

次の例では、fp ポートに接続されたファイバチャネルデバイスを一覧表示しています。

```
# cfgadm -al
     Ap_Id                Type         Receptacle   Occupant     Condition
     c0                   fc-fabric    connected    configured   unknown
     c0::50020f2300006077 disk         connected    configured   unknown
     c0::50020f23000063a9 disk         connected    configured   unknown
     c0::50020f2300005f24 disk         connected    configured   unknown
     c0::50020f2300006107 disk         connected    configured   unknown
     c1                   fc-private   connected    configured   unknown
     c1::220000203708b69c disk         connected    configured   unknown
     c1::220000203708ba7d disk         connected    configured   unknown
     c1::220000203708b8d4 disk         connected    configured   unknown
     c1::220000203708b9b2 disk         connected    configured   unknown
     c2                   fc-pt_to_pt  connected    configured   unknown
     c2::500104f000937528 tape         connected    configured   unknown
     c3                   fc           connected    unconfigured unknown
     sysctrl0:slot0       cpu/mem      connected    configured   ok
     sysctrl0:slot1       sbus-upa     connected    configured   ok
```

この例では、ap_id c0 の fc-fabric タイプが fp ポートがファブリックに接続されていることを示します。ファブリック関連タイプ (fc-fabric や fc-public など) の fp ポートでは、デバイスノードの作成はデフォルトでブート時に発生し、cfgadm による構成操作および構成解除操作で管理できます。ap_id c1 の fc-private タイプは、fp ポートがプライベートループに接続され、デバイスノードの作成も同様にデフォルトで発生することを示します。ap_id c2 の fc-pt_to_pt タイプは、fp ポートが別の N_port に直接接続され、デバイスノードの作成もデフォルトで発生することを示します。ap_id c3 の fc タイプは、fp ポート c2 には何も接続されていないことを示します。ファイバチャネルデバイス ap_id の Type フィールドは、デバイスの LUN 0 の SCSI デバイスタイプを示します。

Solaris には複数の FCP SCSI LUN が搭載されたファイバチャネルデバイスが構成され、各 FCP SCSI LUN は Solaris デバイスとして使用可能です。ap_id c0::50020f2300006077 および c0::50020f23000063a9 は、複数の FCP SCSI LUN が搭載されたファイバチャネルデバイスを表すことを想定します。

次の例では、FCP SCSI LUN 情報を含む ap_id を一覧表示する方法を示します。

```
# cfgadm -al -o show_SCSI_LUN
     Ap_Id                  Type         Receptacle   Occupant     Condition
     c0                     fc-fabric    connected    configured   unknown
     c0::50020f2300006077,0 disk         connected    configured   unknown
     c0::50020f2300006077,1 disk         connected    configured   unknown
     c0::50020f2300006077,2 disk         connected    configured   unknown
     c0::50020f2300006077,3 disk         connected    configured   unknown
     c0::50020f23000063a9,0 disk         connected    configured   unknown
     c0::50020f23000063a9,1 disk         connected    configured   unknown
     c0::50020f23000063a9,2 disk         connected    configured   unknown
     c0::50020f23000063a9,3 disk         connected    configured   unknown
     c0::50020f2300005f24,0 disk         connected    unconfigured unknown
     c0::50020f2300005f24,1 disk         connected    unconfigured unknown
     c0::50020f2300006107,0 disk         connected    unconfigured unknown
     c0::50020f2300006107,1 disk         connected    unconfigured unknown
     c1                     fc-private   connected    configured   unknown
     c1::220000203708b69c,0 disk         connected    configured   unknown
     c1::220000203708ba7d,0 disk         connected    configured   unknown
     c1::220000203708b8d4,0 disk         connected    configured   unknown
     c1::220000203708b9b2,0 disk         connected    configured   unknown
     c2                     fc-pt_to_pt  connected    configured   unknown
     c2::500104f000937528,0 tape         connected    configured   unknown
     c3                     fc           connected    unconfigured unknown
```

この例では、ap_id c0::50020f2300006077,0 は、ポート WWN 50020f2300006077 で表現されるファイバチャンネルデバイスの FCP SCSI LUN 0 を識別します。ファイバチャンネルデバイスは 4 つの FCP SCSI LUN が搭載されていると報告され、すべて構成済みです。ポート WWN 50020f2300006077 に関連付けられた 4 FCP SCSI LUN レベルの ap_id が一覧表示されます。一覧には、未構成のファイバチャネルデバイスの FCP SCSI LUN も表示されます。c0::50020f2300005f24 で表現されるファイバチャネルデバイスは、2 つの FCP SCSI LUN が搭載されていると報告されます。c0::50020f2300005f24 での構成操作では、2 つの Solaris デバイスが作成されます。FCP SCSI LUN レベル ap_id の Type フィールドには、各 LUN の SCSI デバイスタイプが表示されます。ファイバチャネルデバイスに別のタイプの LUN がある場合は、Type フィールドに反映されます。

fp ポート上の接続点の受容体と占有装置の状態は、次の意味を持ちます。

configured
: 1 つ以上のデバイスが fp ポートに構成されている

connected
: fp ポートがアクティブ

disconnected
: fp ポートが休止 (入出力アクティビティーが停止)

empty
: 該当しない

unconfigured
: fp ポートにデバイスの構成なし

fp ポートでの各ファイバチャネルデバイスの状態は次のとおりです。

configured
: デバイスが Solaris に構成され、使用可能

connected
: デバイスが接続される fp ポートがアクティブ

disconnected
: デバイスが接続される fp ポートが休止

unconfigured
: デバイスが構成可能

fp ポートにおける接続点の condition フィールドは、次のような意味を持ちます。

failed
: エラー状態のため、fp ポートがファイバチャネル接続の存在やタイプを検出できませんでした。

fp ポートにおける各ファイバチャネルデバイスの condition フィールドは、次のような意味を持ちます。

failed
: ファブリック上のデバイスの検索中にエラーが発生しました。

failing
: デバイスはホスト上に構成され、その状態は Solaris で認識されているとおり通常 (オンラインなど) であると表示されますが、現在ファブリックに存在しない、または表示されていないか、デバイスが構成されたローカルポートがエラー状況であるために存在を確認できませんでした。

unusable
: デバイスはホスト上に構成されていますが、現在オフラインであるか、障害が発生しています。

不明な「状況」は、ファブリック上のデバイスの検索がエラーなしで完了し、デバイスが事前に構成された場合の Solaris ホスト内のデバイス状態は通常です。デバイスの内部状況は保証できません。

オプション

cfgadm では、一覧表示 (-l) のほかに数種類の操作が定義されています。これらの操作には、構成状態の変更の開始や構成管理のヘルプメッセージの取得 (-h) が含まれます。

次のオプションがサポートされています。

-c *function*

fp トランスポート固有ライブラリでは、次の汎用コマンドが定義されています。

fc-fabric タイプ fp ポート接続点上のファイバチャネルデバイス接続点では、次の構成状態の変更操作がサポートされています。

configure

接続されたファイバチャネルファブリックデバイスをホストに構成します。ファイバチャネルデバイスが一覧操作の出力に不明なタイプとして一覧表示されている場合は、デバイスが構成できない可能性があります。不明なタイプのデバイスの構成は試みられません。強制オプション (-f) を使用すると、任意のデバイスの構成を試みるように fp ポートドライバプラグインを強制できます。プロセスのエラーが報告されます。デフォルトでは、ファイバチャネルファブリックデバイスで検出された各 FCP SCSI LUN が構成されます。ただし、fp.conf ファイルの「pwwn-lun-blacklist」プロパティーで指定された FCP SCSI LUN は構成解除されたままです。FCP SCSI LUN レベル一覧には、このような FCP SCSI LUN の状態が反映されます。接続されたコントローラでリブートまたは Solaris 動的再構成を実行した後も、「構成解除」の状態が保持されます。「pwwn-lun-blacklist」プロパティーについての詳細は、fp(7d) を参照してください。

unconfigure

ファイバチャネルファブリックデバイスをホストから構成解除します。デバイスが接続されたコントローラで次のリブートまたは Solaris 動的再構成が実行されるまで、このデバイスは構成解除されたままです。この時点で、すべてのファブリックデバイスが自動的に列挙されます。デフォルト動作は、fp.conf ファイルの「manual_configuration_only」プロパティーを使用することによって変更できます。このプロパティーが設定されている場合、リブート後もデバイスは構成解除されたままです。「manual_configuration_only」プロパティーについての詳細は、fp(7d) を参照してください。

ファイバチャネルのプライベートループデバイスおよび N_Port ポイントツーポイントデバイスでは、configure コマンドは操作を行うことなく正常終了を返します。プライベートループデバイスおよび N_Port ポイントツーポイントデバイスでは、unconfigure コマンドがサポートされていません。プライベートループデバイスおよび N_Port ポイントツーポイントデバイスは、デフォルトで Solaris ファイバチャネルドライバによって構成され、エンドユーザーまたはアプリケーション主導の操作によって管理されません。fp.conf ファイルの pwwn-lun-blacklist プロパティーは、ファブリックデバイスに適用する場合と同じ方法でプライベートループデバイスおよび N_Port ポイントツーポイントデバイスに適用されます。

-f
状況やタイプに関係なく、configure による状態の変更操作を強制的に発生させます。configure による状態の変更操作については、上記の説明を参照してください。

-h *ap_id*
fp (トランスポート固有のヘルプ) を取得します。任意の fp 接続点を指定します。

-o*hardware_options*
次のハードウェアオプションがサポートされています。

show_SCSI_LUN
list オプション -al を付けて指定した場合、検出されたファイバチャネルデバイスについて各 FCP SCSI LUN に関連付けられた ap_id を一覧表示します。前述の説明および FCP SCSI LUN レベル一覧の例を参照してください。FCP SCSI LUN レベルでは、デバイスノードの作成はサポートされていません。「注意事項」 を参照してください。

Solaris では、すべてのファイバチャネルデバイスがデフォルトで使用可能です。/kernel/drv/fp.conf ファイルの「manual_configuration_only」プロパティーを設定すると、Solaris でファブリックデバイスのサブセットのみをデフォルトで使用可能にできます。fp.conf の「manual_configuration_only」を設定すると、アプリケーションまたはエンドユーザーが Solaris にデバイスが構成されるように事前に要求していない限り、Solaris ですべてのファブリックデバイスが使用不可になります。configure による状態の変更コマンドは、デバイスを Solaris で使用可能にします。ファブリックデバイスで configure 操作が正常に完了すると、関連するリンクが /dev 名前空間に追加されます。unconfigure による状態の変更コマンドは、デバイスを Solaris で使用不可にします。

-c configure 操作を使用してファイバチャネルファブリックデバイスがホストに正常に構成されると、物理 ap_id がリポジトリに格納されます。-c unconfigure 操作を使用してファイバチャネルファブリックデバイスが構成解除されると、物理 ap_id が同じリポジトリから削除されます。fp.conf の「manual_configuration_only」プロパティーが設定されている場合にのみ、すべてのファブリックデバイスがデフォルトで自動的に列挙され、リポジトリが使用されます。「manual_configuration_only」プロパティーについての詳細は、fp(7d) を参照してください。

次のコマンドを -c オプションを付けて指定すると、リポジトリの更新動作を制御できます。

force_update
configure では、接続点が無条件にリポジトリに追加されます。unconfigure では、接続点が無条件に削除されます。

no_update
操作に関係なく、リポジトリの更新は実行されません。

これらのオプションは、通常の configure および unconfigure 操作では使用してはいけません。「警告」を参照してください。

ファイバチャネルデバイスに複数の FCP SCSI LUN が構成され、FCP SCSI LUN に関連付けられている任意の Solaris デバイスが使用不可の状況である場合は、ファイバチャネルデバイス全体が使用不可として報告されます。-c unconfigure コマンドに次のオプションを付けると、ファイバチャネルデバイスが使用不可の状況である Solaris デバイスのみが削除されます。

unusable_SCSI_LUN
unconfigure 操作では、ターゲットデバイスでオフラインのデバイスノードが削除されます。

-s*listing_options*
使用法については、cfgadm(1M) を参照してください。

-t *ap_id*
現時点では、使用できるテストコマンドはありません。

-x*hardware_function*
現時点では、ハードウェア固有の関数は使用できません。

その他のすべてのオプションの意味は、cfgadm(1M) のマニュアルページで定義されたものと同じです。

使用例

例 1 ディスクの構成解除

次のコマンドは、ディスクを構成解除します。

```
# cfgadm -c unconfigure c0::210000203708b606
```

例 2 単一の接続点にあるすべての構成済みディスクの構成解除

次のコマンドは、接続点 c0 にあるすべての構成済みディスクを構成解除します。

```
# cfgadm -c unconfigure c0
```

例 3 ディスクの構成

次のコマンドは、ディスクを構成します。

```
# cfgadm -c configure c0::210000203708b606
```

例 4 単一の接続点にあるすべての構成解除済みディスクの構成

次のコマンドは、接続点 c0 にあるすべての構成解除済みディスクを構成します。

```
# cfgadm -c configure c0
```

例 5 リポジトリからのファイバチャネルファブリックデバイスの接続点の削除

次のコマンドは、ファブリックデバイスリポジトリからファイバチャネルファブリックデバイスの接続点を無条件に削除します。

例 5 リポジトリからのファイバチャネルファブリックデバイスの接続点の削除 (続き)

```
# cfgadm -c unconfigure -o force_update c0::210000203708b606
```

例 6 ターゲットデバイスでオフラインの Solaris デバイスノードの削除

次のコマンドは、ターゲットデバイスでオフラインの Solaris デバイスノードを削除します。

```
# cfgadm -c unconfigure -o unusable_SCSI_LUN c0::210000203708b606
```

ファイル

`/usr/lib/cfgadm/fp.so.1`
: ファイバチャネルファブリックのデバイスノード管理用のハードウェア固有ライブラリ。

`/etc/cfg/fp/fabric_WWN_map`
: 現在構成されているファブリックデバイスの物理 `ap_id` のリポジトリ。ブート時にこれらのファブリックデバイスを再構成する場合にのみ使用されます。このリポジトリは、`/kernel/drv/fp.conf` ファイルの「manual_configuration_only」が設定されている場合にのみ使用されます。

`/etc/rcS.d/fdevattach`
: ブート時に `/etc/cfg/fp/fabric_WWN_map` に一覧表示される物理 `ap_id` のファブリックデバイスを再構成します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	service/storage/fibre-channel/fc-fabric, service/storage/fibre-channel/fc-fabricx

関連項目

`svcs(1)`, `cfgadm(1M)`, `svcadm(1M)`, `config_admin(3CFGADM)`, `libcfgadm(3LIB)`, `attributes(5)`, `smf(5)`, `fp(7d)`

警告

通常の `configure/unconfigure` 操作では、リポジトリ更新用のハードウェア固有オプションを使用しないでください。ブート時にファブリックデバイスのノード作成に失敗して、エラー状態が永続的であると見なされるとき、ハードウェア固有のオプションが使用されると想定されます。`unconfigure` コマンドに `force_update` ハードウェア固有オプションを付けると、障害が発生しているファブリックデバイスの接続点がリポジトリから無条件に削除されます。

注意事項

不明なデバイスタイプまたは SCSI デバイスタイプ以外のデバイス (ファイバチャネルホストバスアダプタなど) では、`configure` 操作を適用できない場合があります。

`configure` および `unconfigure` コマンドは、ポート WWN ap_id で表現されるファイバチャネルデバイスレベルで動作します。ファイバチャネルデバイスに複数の FCP SCSI LUN が構成されている場合、関連するポート WWN ap_id で `configure` コマン

ドを実行すると、fp.conf ファイルの「pwwn-lun-blacklist」プロパティーに指定されていない限り、各 FCP SCSI LUN に Solaris デバイスが作成されます。unconfigure コマンドは、ポート WWN ap_id に関連付けられたすべての Solaris デバイスを削除します。configure および unconfigure コマンドでは、FCP SCSI LUN レベル ap_id は無効です。

非推奨の show_FCP_dev オプションは新しい show_SCSI_LUN オプションに置き換えられ、非推奨の unusable_FCP_dev オプションは新しい unusable_SCSI_LUN オプションに置き換えられました。

cfgadm_fp サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/device/fc-fabric:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

このサービスがブート時に開始されれば、ファブリックデバイスを構成するときに、このサービスに対する管理操作が必要ありません。

名前

cfgadm_ib – cfgadm の InfiniBand ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm -f [-y | -n] [-v] -c function ap_id...

/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options]
     -x hardware_function ap_id...

/usr/sbin/cfgadm -v [-a] [-s listing_option] [-] [ap_id | ap_type...]

/usr/sbin/cfgadm -v -h [ap_id]...
```

機能説明

InfiniBand ハードウェア固有のライブラリ `/usr/lib/cfgadm/ib.so.1` は、`cfgadm(1M)` ユーティリティーを使用してファブリックを管理する機能を提供します。`cfgadm` は、接続点で実行されます。`cfgadm(1M)` を参照してください。

InfiniBand (IB) デバイスは、IB デバイスマネージャー (IBDM)に基づいて IB 連結ドライバ `ib(7D)` によって列挙されます。

IB 連結ドライバは、5 種類の子デバイスノードを作成して初期化します。

- IB Port デバイス
- IB HCA サービス (HCA_SVC) デバイス
- IB 仮想物理接続点 (VPPA) デバイス
- 入出力コントローラ (IOC)
- IB 擬似デバイス

IB Port、IB VPPA、および IB HCA_SVC デバイスの列挙についての詳細は、`ib(7D)` を参照してください。IBDM についての追加情報は、`ibdm(7D)` を参照してください。IB 擬似デバイスについての詳細は、`ib(4)` を参照してください。

IB 管理では、特定のホストで認識されるように、ファブリック管理用に 2 種類の静的接続点が作成されます。静的接続点 `ib` が 1 つあり、ファブリックの IB デバイス (IOC、ポート、VPPA、HCA_SVC、または擬似デバイス) はすべてここから分離される動的接続点として表現されます。ホストの各ホストチャネルアダプタ (HCA) には、そのノードのグローバル一意識別子 (GUID) 値に基づく別の静的接続点があります。

接続点には、ap_id を通じて名前が付けられます。論理 ap_id と物理 ap_id の 2 種類の ap_id が定義されます。物理 ap_id は、物理パス名に基づきます。IB ファブリックの場合、`/devices/ib:fabric` です。論理 ap_id は、より短く、ユーザーにわかりやすい名前を持ちます。

IB ファブリックの静的 ap_id は `ib` です。IB デバイスは動的接続点であり、物理 ap_id はありません。IOC の論理 ap_id に含まれる GUID は `ib::`*IOC-GUID* です。IOC ap_id の例は `ib::80020123456789a` です。擬似デバイスの論理 ap_id (詳細は ib(4) を参照) の形式は `ib::`*driver_name*`,`*unit-address* です。擬似 ap_id の例は `ib::sdp,0` です。ここで、「`sdp`」はドライバ名、「`0`」はその *unit-address* プロパティーです。ポート、VPPA、および HCA_SVC デバイスの論理 ap_id には、パーティションキー (*P_Key*)、*Port GUID* / *Node GUID*、および通信に関する service-name が含まれます。ap_id の形式は次のとおりです。

ポートデバイス
ib::*PORT_GUID*,*0*,*service-name*

VPPA デバイス
ib::*PORT_GUID*,*P_Key*,*service-name*

HCA_SVC デバイス
ib::*HCA_GUID*,0,*servicename*

ポートデバイスおよび HCA_SVC デバイスの場合、パーティションキー (*P_Key*) は 0 です。*P_Key* は、VPPA デバイスノード用にこのポートが属するパーティションを特定する際に役立ちます。ポートには複数の *P_Key* がある場合があります。VPPA デバイスの論理 ap_id ポイントの例は ib::80245678,ffff,ipib です。*port-GUID* は 80245678、*P_Key* は 0xffff、サービス名は ipib です。service-name 情報は、service-name 文字列が含まれるファイル /kernel/drv/ib.conf から取得されます。HCA の論理 ap_id に含まれるノード GUID 値は hca:HCA-GUID です。例は hca:21346543210a987 です。

IB 接続点の一覧には、ホストで参照されず、使用されるように構成されていない場合でも、ファブリック内のすべての IB デバイス (IBDM および IBIOC 連結ドライバで参照される VPPA、HCA_SVC、擬似、およびポートデバイス) に関する情報が含まれます。

次の一覧は、5 つの IB デバイス (2 つの IOC、1 つの VPPA、1 つのポート、1 つの HCA_SVC) と 1 つの HCA を示しています。

```
example# cfgadm -al
Ap_Id                     Type       Receptacle   Occupant      Condition
hca:21346543210a987       IB-HCA     connected    configured    ok
ib                        IB-FABRIC  connected    configured    ok
ib::80020123456789a       IB-IOC     connected    configured    ok
ib::802abc9876543         IB-IOC     connected    unconfigured  unknown
ib::80245678,ffff,ipib    IB-VPPA    connected    configured    ok
ib::12245678,0,nfs        IB-PORT    connected    configured    ok
ib::21346543,0,hnfs       IB-HCA_SVC connected    configured    ok
ib::sdp,0                 IB-PSEUDO  connected    configured    ok
```

ap_id ib::802abc9876543 は、ホストで使用されるように構成されていない、または明示的な操作によって事前にオフラインにされていない IOC デバイスを

```
cfgadm -c unconfigure
```

示しています。Condition 列に表示される情報で区別されます。P_Key と HCA GUID がゼロの IB デバイスは HCA_SVC デバイスです。接続点の一覧表示については、cfgadm(1M) を参照してください。

接続点の受容体の状態は、次の意味を持ちます。

connected
: IOC/VPPA/Port/Pseudo/HCA_SVC デバイスでは、`connected` はホストで参照されていることを意味します。デバイスは Solaris で使用されるように構成されてない場合もあります。

  HCA 接続点では、`connected` はデバイスが構成されていて、使用中であることを意味します。

  すべての IB ap_id は、常に connected と表示されます。

接続点の占有装置の状態は、次の意味を持ちます。

configured
: IB デバイスおよび HCA ap_id が構成され、Solaris で使用可能です。

unconfigured
: ap_id の IB デバイスは `cfgadm -c unconfigure` を使用して明示的にオフラインにされましたが、正常に構成されませんでした。これは、デバイスが Solaris で使用されるように正常に構成されなかったか (ドライバがない、またはデバイスの問題のため)、IB 連結ドライバで使用されるように構成されなかったことが原因になっている可能性があります。

  HCA 接続点では、`unconfigured` 操作はサポートされていません。システムに IB ハードウェアがない場合は、IB の静的 apid (`ib`) は unconfigured と表示されます。

接続点の条件は次のとおりです。

failed
: 使用されていません。

failing
: 使用されていません。

ok
: 通常の状態です。使用できます。

unknown
: この状態は、IBDM で検索されたものの、Solaris で使用されるように構成されていない IB デバイスでのみ有効です。また、`cfgadm -c unconfigure` 操作で明示的にオフラインになったデバイスにも表示されます。この状況は、HCA 接続点には適用されません。

unusable
: 使用されていません。

オプション

次のオプションがサポートされています。

-c *function*

IB ハードウェア固有のライブラリでは、2 つの汎用コマンド (関数) がサポートされています。これらのコマンドは、静的接続点 (つまり、HCA ap_id および IB の静的 ib ap_id) ではサポートされていません。

次の汎用コマンドがサポートされています。

configure

Solaris で使用されるように IB デバイスを構成します。

unconfigure

IB デバイスの構成を解除します。正常に完了した場合、cfgadm ではこの ap_id の状況が unknown と報告されます。

-f

サポートされていません。

-h *ap_id*

IB 接続点に関する IB 固有のヘルプを取得します。

-l

IB 接続点の状態および状況を一覧表示します。-l オプションは、cfgadm(1M) で説明されているように機能します。

-a オプションとともに使用すると、動的接続点 (IOC、VPPA、ポート、擬似、および HCA_SVC デバイス) も同様に表示されます。

-v オプションとともに使用すると、ap_id に関する詳細データが表示されます。IOC の場合、出力の Info フィールドには、

```
cfgadm -avl
```

IOCControllerProfile にある VendorID、IOCDeviceID、DeviceVersion、SubsystemVendorID、SubsystemID、Class、Sub よび IDString に関する情報が表示されます。ID 文字列が指定されていない場合、その場所には何も表示されません。これらのフィールドは、『InfiniBand Specification Volume 1』(http://www.infinibandta.org) に定義されています。

VPPA、ポート、または HCA_SVC デバイスの場合、cfgadm -lav の Info フィールドには、このデバイスがバインドされたサービス名に関する情報が表示されます。このような情報が存在しない場合は、何も表示されません。

擬似デバイス cfgadm -alv の場合、ドライバ名とその unit-address 情報が表示されます。HCA の場合、詳細一覧には VendorID、HCA の ProductID、ポートの数、およびそのポートの PortGUID 値が表示されます。「使用例」を参照してください。

-o *hardware_option*

このオプションは、現在定義されていません。

-s *listing_option*
select サブオプションを使用すると、クラス ib の接続点を一覧表示できます。詳細は、cfgadm(1M) のマニュアルページを参照してください。

-x*hardware_function*
ハードウェア固有の関数を実行します。*name* は 4 文字よりも長くできないことに注意してください。

次のハードウェア固有の関数がサポートされています。

add_service -ocomm=[port|vppa|hca_svc],service=*name*
このハードウェア固有の関数は、静的 IB 接続点でサポートされています。これを使用すると、新しいサービスを /kernel/drv/ib.conf ファイルに追加したり、ib(7D) ドライバを更新したりできます。

新しいサービスが追加されるように指定するには、service=*name* オプションを使用する必要があります。name サービスを /kernel/drv/ib.conf ファイルの port-svc-list または hca-svc-list に追加するには、comm=[port|vppa|hca_svc] オプションを使用する必要があります。「使用例」を参照してください。

delete_service -ocomm=[port|vppa|hca_svc],service=*name*
このハードウェア固有の関数は、静的 IB 接続点でのみサポートされています。これを使用すると、既存のサービスを /kernel/drv/ib.conf ファイルから削除したり、ib(7D) ドライバのデータベースから削除したりできます。削除するサービスを指定するには、service=name オプションを使用する必要があります。このサービスを /kernel/drv/ib.conf ファイルの port-svc-list、vppa-svc-list、または vppa-svc-list から削除するには、comm=[port|vppa|hca_svc] オプションを使用する必要があります。「使用例」を参照してください。

list_clients
HCA 接続点でサポートされています。この HCA を使用して、すべてのカーネル IB クライアントを表示します。また、これらのカーネル IB クライアントおよび代替 HCA デバイス (開いている場合) の各 ap_id も表示されます。「使用例」を参照してください。

.

特定のカーネル IB クライアントに有効な ap_id がない場合は、その列に - が表示されます。

list_services
このハードウェア固有の関数は、静的 IB 接続点でのみサポートされています。/kernel/drv/ib.conf ファイルから読み取られるすべてのポートおよび VPPA サービスが一覧表示されます。「使用例」を参照してください。

unconfig_clients
このハードウェア固有の関数は、静的 HCA 接続点でのみサポートされています。これを使用すると、この特定 HCA のすべてのカーネル IB クライアントを構成解除できます。代替 HCA がないカーネル IB クライアントのみが構成解除されます。「使用例」を参照してください。

update_ioc_config
このハードウェア固有の関数は、静的 IB 接続点および IOC 接続点でサポートされています。ib APID の場合、この関数はすべての IOC デバイスノードのプロパティーを更新します。IOC APID の場合、この関数は指定された IOC デバイスノードのプロパティーを更新します。このコマンドは port-list、port-entries、service-id、および service-name IOC ノードプロパティーを更新します。

ib(7D) を参照してください。

update_pkey_tbls
静的 ib 接続点でサポートされています。IBTL 内部の PKEY 情報を更新します。IBTL は、ホストに存在する各 HCA 上のすべてのポートについて P_Key テーブルを再度読み取ります。

ibtl(7D) を参照してください。

使用例

例 1 IB デバイスの状態および状況の一覧表示

次のコマンドは、システム上の IB デバイスの状態および状況を一覧表示します。静的接続点のみが表示されます。

```
example# cfgadm
hca:21346543210a987        IB-HCA      connected    configured   ok
ib                         IB-FABRIC   connected    configured   ok
```

-a オプションは、すべての接続点を表示します。次の例では、-a オプションを使用して、すべての接続点を一覧表示します。

```
example# cfgadm -a
hca:21346543210a987          IB-HCA      connected    configured   ok
ib                           IB-FABRIC   connected    configured   ok
ib::80020123456789a          IB-IOC      connected    unconfigured ok
ib::80245678,ffff,ipib       IB-VPPA     connected    configured   ok
ib::21346543,0,hnfs          IB-HCA_SVC  connected    configured   ok
ib::12245678,0,nfs           IB-PORT     connected    configured   ok
ib::sdp,0                    IB-PSEUDO   connected    configured   ok
```

例 2 IB VPPA デバイスの詳細ステータスの一覧表示

次のコマンドは、IB VPPA デバイスの詳細ステータスを一覧表示します。

```
example# cfgadm -alv ib::80245678,ffff,ipib
Ap_Id                    Receptacle Occupant   Condition Information
```

例 2　IB VPPA デバイスの詳細ステータスの一覧表示　　(続き)

```
When         Type     Busy  Phys_Id
ib::80245678,ffff,ipib    connected    configured   ok        ipib
unavailable  IB-VPPA  n     /devices/ib:fabric::80245678,ffff,ipib
```

IOC の詳細一覧には、追加情報が表示されます。次のコマンドは、詳細一覧を表示します。

```
example# cfgadm -alv ib::80020123456789a
Ap_Id      Receptacle   Occupant     Condition Information
When       Type     Busy  Phys_Id
ib::80020123456789a     connected    configured   ok         VID: 0xeaea
DEVID: 0xeaea VER: 0x5 SUBSYS_VID: 0x0 SUBSYS_ID: 0x0 CLASS: 0xffff
SUBCLASS: 0xff PROTO: 0xff PROTOVER: 0x1 ID_STRING: Sample Host Adapter
unavailable IB-IOC   n     /devices/ib:fabric::80020123456789a
```

擬似デバイスの詳細一覧は、次のものを表示します。

```
example# cfgadm -alv ib::sdp,0
Ap_Id                 Receptacle  Occupant   Condition Information
When       Type   Busy  Phys_Id
ib::sdp,0         connected   configured   ok       Driver = "sd
p" Unit-address = "0"
unavailable  IB-PSEUDO   n  /devices/ib:fabric::sdp,0
```

HCA の詳細一覧は、次のものを表示します。

```
example# cfgadm -alv hca:21346543210a987
Ap_Id               Receptacle   Occupant     Condition Information
When       Type    Busy  Phys_Id
hca:21346543210a987  connected    configured   ok         VID: 0x15b3,
PID: 0x5a44, #ports: 0x2, port1 GUID: 0x80245678, port2 GUID: 0x80245679
unavailable  IB-HCA     n  /devices/ib:21346543210a987
```

このような `cfgadm` クラスおよびフィールドの selection オプション (`-s "select=class(ib),cols=ap_id:info"`) を指定すると、ユーザーにわかりやすい出力を取得できます。

次のコマンドは、IB ap_id のみを表示します。出力には、`ap_id` および `Information` フィールドのみが含まれます。

```
# cfgadm -al -s "cols=ap_id:info"  ib::80245678,ffff,ipib
Ap_Id                                Information
ib::80245678,ffff,ipib               ipib
```

例 3　既存の IB IOC の構成解除

次のコマンドは、`ib::80020123456789a` に接続された IB IOC を構成解除してから、`ap_id` のステータスを表示します。

例 3　既存の IB IOC の構成解除　　(続き)

```
# cfgadm -c unconfigure ib::80020123456789a
Unconfigure the device: /devices/ib:fabric::80020123456789a
This operation will suspend activity on the IB device
Continue (yes/no)?
```

「y」を入力します。

```
IB device unconfigured successfully.
# cfgadm -al ib::80020123456789a
Ap_Id                  Type      Receptacle  Occupant     Condition
ib::80020123456789     IB-IOC    connected   unconfigured unknown
#
```

状況 unknown は、デバイスノードがすでに存在せず、この IB デバイスの存在が IB デバイスマネージャーにしか認識されていないことを意味します。

例 4　IB IOC の構成

次の一連のコマンドは、ib::80020123456789a に接続された IB デバイスを構成します。

```
# cfgadm -yc configure ib::80020123456789a
# cfgadm -al ib::80020123456789a
Ap_Id                  Type        Receptacle   Occupant     Condition
ib::80020123456789a    IB-IOC      connected    configured   ok
```

例 5　HCA のすべてのカーネル IB クライアントの一覧表示

次のコマンドは、hca:21346543210a987 に接続された HCA のすべてのカーネル IB クライアントを一覧表示します。

```
# cfgadm -x list_clients hca:21346543210a987
Attachment Point       Clients                Alternate HCA
ib::80020123456789a    ioc1                   Yes
ib::80245678,ffff,ipib ipib                   No
ib::21346543,0,hnfs    hnfs                   No
-                      ibdm                   No
-                      ibmf                   No
```

例 6　ポートサービスの追加

次のコマンドは、srp という新しいポートサービスを追加します。

```
 # cfgadm -o comm=port,service=srp -x add_service ib
```

例 7　VPPA サービスの削除

次のコマンドは、ibd VPPA サービス ibd を削除します。

```
# cfgadm -o comm=vppa,service=ipib -x delete_service ib
```

例 8　ポート、VPPA、HCA_SVC サービスの一覧表示

次のコマンドは、すべてのポート、VPPA、および HCA_SVC サービスを一覧表示します。

```
# cfgadm -x list_services ib
Port communication services:
         srp

VPPA communication services:
         ipib
         nfs

HCA_SVC communication services:
         hnfs
```

例 9　IOC デバイスの再検索

次のコマンドは、すべての IOC デバイスノードを再検索します。

```
# cfgadm -x update_ioc_config ib
This operation can update properties of IOC devices.
Continue (yes/no)?

Enter: y

#
```

例 10　HCA のすべてのカーネルクライアントの構成解除

次のコマンドは、HCA のすべてのカーネルクライアントを構成解除します。

```
# cfgadm -x unconfig_clients hca:21346543
 This operation will unconfigure clients of this HCA.
 Continue (yes/no)?

 Enter: y
```

ファイル

`/usr/lib/cfgadm/ib.so.1`
　汎用 InfiniBand デバイス管理のためのハードウェア固有ライブラリ

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

`cfgadm(1M)`, `config_admin(3CFGADM)`, `libcfgadm(3LIB)`, `ib(4)`, `attributes(5)`, `ib(7D)`, `ibdm(7D)`, `ibtl(7D)`

InfiniBand Specification Volume 1 (http://www.infinibandta.org)

注意事項

一覧表示 (`cfgadm -l` または `cfgadm -x list_clients`) を除いて、接続点ではスーパーユーザーのみが関数を実行できます。

名前

cfgadm_pci – cfgadm の PCI、CompactPCI、および PCI Express ホットプラグ対応ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v]
     [-o hardware_options] -c function ap_id [ap_id]

/usr/sbin/cfgadm [-f] [-y | -n] [-v]
     [-o hardware_options] -x hardware_function ap_id
     [ap_id]

/usr/sbin/cfgadm [-v] [-s listing_options]
     [-o hardware_options] [-l [ap_id | ap_type]]

/usr/sbin/cfgadm [-v] [-o harware_options] -t ap_id [ap_id]

/usr/sbin/cfgadm [-v] [-o hardware_function] -h
     [ap_id| ap_type]
```

機能説明

PCI ハードウェア固有のライブラリ `/usr/lib/cfgadm/pci.so.1` では、`cfgadm` コマンド (`cfgadm(1M)` を参照) を使用した、ホットプラグ対応システムの各ホットプラグ対応スロットへの PCI および CompactPCI アダプタカードのホットプラグ操作がサポートされています。このライブラリでは、別のライブラリで提供される PCI Express ホットプラグ対応または標準 PCI ホットプラグ対応アダプタカード (`cfgadm_shp(1M)` を参照) はサポートされません。このドキュメントに示されている点を除き、PCI と CompactPCI 間のホットプラグ管理モデルは同じです。

PCI ホットプラグの場合、特定の PCI バス上の各ホットプラグスロットは、その特定の PCI バスの接続点によって表されます。

接続点は、受容体と占有装置の 2 つの部分で構成されます。通常、PCI ホットプラグにおける受容体は、物理ホットプラグ対応スロットと見なされ、占有装置はスロットに接続される PCI アダプタカードと見なされます。

接続点には、`ap_id` を通じて名前が付けられます。`ap_id` には、論理 ap_id と物理 ap_id の 2 種類があります。物理 `ap_id` は物理パス名 (つまり、`/devices/pci@1/hpc0_slot3`) に基づきます。一方、論理 `ap_id` はより短く、ユーザーにわかりやすい名前です。PCI ホットプラグ対応スロットの場合、通常 `ap_id` は対応するホットプラグコントローラドライバ名に論理スロット番号を加えたものです。つまり、`pci0:hpc0slot1` は、`hpc` というホットプラグコントローラドライバとスロット番号 1 を含む PCI 連結ドライバです。PCI ホットプラグの `ap_type` は `pci` です。

`ap_type` は、`Type` フィールドの情報と同じではない点に注意してください。

ホットプラグプロシージャについての詳細は、『Oracle Solaris の管理: 一般的なタスク』を参照してください。

オプション

次のオプションがサポートされています。

-c *function*
PCI ホットプラグ対応スロットでは、次の関数がサポートされています。

configure
スロット上の PCI デバイスが Solaris で使用されるように構成します。

connect
スロットを PCI バスに接続します。

disconnect
PCI バスからスロットを切断します。

insert
サポートされていません。

remove
サポートされていません。

unconfigure
PCI デバイスのリソースをシステムから論理的に削除します。

-f
サポートされていません。

-h *ap_id* | *ap_type*
PCI ホットプラグ固有のヘルプメッセージを出力します。

-l *list*
PCI ホットプラグスロットの値を一覧表示します。

-o*hardware_options*
現在定義されているハードウェア固有のオプションはありません。

-s*listing_options*
汎用の cfgadm(1M) と同じです。

-t *ap_id*
このコマンドは、スロットでテスト機能をサポートするプラットフォームでのみサポートされます。

-v
詳細モードで実行します。

-v オプションを -l オプションとともに使用すると、cfgadm コマンドによって、接続点に関する情報が出力されます。PCI Express 階層内に存在する PCI ホットプラグの接続点については、cfgadm_shp(1M) を参照してください。PCI Express 階層内に存在しない PCI ホットプラグの接続点については、スロットのシステムラベルがあれば Information フィールドに表示されます。この文字列は、スロットのバスノードの slot-name プロパティーから取得されます。Type

フィールドの情報は、v オプションを指定したかどうかにかかわらず出力されます。占有装置の Type フィールドは、スロットの内容を示します。次の 2 種類の値があります。

unknown
: スロットは空です。スロットにカードが装着されている場合、カードが構成されていないか、カード上のデバイスのドライバがありません。

subclass/*board*
: スロットに装着されているカードは、シングルファンクションデバイスまたはマルチファンクションデバイスです。

  subclass はデバイスのサブクラスコードを表す文字列であり、SCSI、ethernet、pci-isa などが示されます。カードがマルチファンクションデバイスである場合、代わりに MULT が出力されます。

  board は、デバイスのボードタイプを表す文字列です。たとえば、hp は PCI ホットプラグアダプタ、hs はホットスワップボード、nhs は非ホットスワップ cPCI ボード、bhs はベーシックホットスワップ cPCI ボード、fhs はフルホットスワップ cPCI ボードで使用される文字列です。

  複数のデバイスが搭載された PCI カードはマルチファンクションデバイスではありませんが、背後に任意のデバイスが搭載された PCI ブリッジとして実装されます。このような場合、表示されるサブクラスはその PCI ブリッジのものです。一般には、ブリッジは pci-pci (PCI ブリッジへの汎用 PCI) または stpci (半透過性 PCI ブリッジ) です。

-x*hardware_function*
: ハードウェア固有の関数を実行します。通常は、これらのハードウェア固有の機能を使用して、受容体または占有装置の状態を変更しないでください。

  次の *hardware_functions* がサポートされています。

  enable_slot | disable_slot
  : スロットの状態を変更し、リブート後もスロットの状態を保持します。リブート後の状態の保持は、一部のプラットフォームでのみサポートされます。

    enable_slot は、ホットプラグイン用およびブート時に、このスロットへのハードウェアの追加を有効にします。

    disable_slot は、ホットプラグイン用およびブート時に、このスロットへのハードウェア追加を無効にします。スロットを無効にすると、状態は unusable と表示されます。

  enable_autoconfig | disable_autoconfig
  : スロットの占有装置を自動構成する機能を変更します。自動構成がサポートされているプラットフォームのみが、この機能をサポートします。

`enable_autoconfig` は、スロットを自動構成する機能を有効にします。

`diable_autoconfig` は、スロットを自動構成する機能を無効にします。

自動構成は、PCI Express プラットフォームのアテンションボタンおよび CompactPCI プラットフォームの取り付け/取り外しレバーから実行されます。自動構成を無効にすると、アテンションボタンまたはレバーメカニズムを使用してスロットの占有装置を構成できなくなります。

led=[*led_sub_arg*],mode=[*mode_sub_arg*]
サブ引数を指定しない場合、現在の LED 設定の一覧を出力します。サブ引数を指定した場合、スロットの特定の LED のモードが設定されます。

led_sub_arg は、`fault`、`power`、`attn`、または `active` に指定します。

mode_sub_arg は、`on`、`off`、または `blink` に指定します。

LED の状態を変更しても、受容体または占有装置の状態は変更されません。通常は、LED はホットプラグコントローラによって制御され、ユーザーの操作は不要です。このコマンドは、テストのために使用します。

注意: LED の状態を変更すると、占有装置または受容体の状態が誤って示される場合があります。

次のコマンドは、LED の値を出力します。

```
example#  cfgadm -x led pci0:hpc0_slot1
Ap_Id             Led
pci0:hpc0_slot1   power=on,fault=off,active=off,attn=off
```

次のコマンドは、Fault LED をオンに設定します。

```
example# cfgadm -x led=fault,mode=on pci0:hpc0_slot1
```

次のコマンドは、Power LED をオフに設定します。

```
example# cfgadm -x led=power,mode=off pci0:hpc0_slot0
```

次のコマンドは、アクティブな LED が点滅してスロットの場所を示すように設定します。

```
example# cfgadm -x led=active,mode=on pci0:hpc0_slot3
```

使用例

例 1 各スロットの値の出力

次のコマンドは、各スロットの値を出力します。

```
example# cfgadm -l
Ap_Id            Type         Receptacle   Occupant       Condition
c0               scsi-bus     connected    configured     unknown
c1               scsi-bus     connected    unconfigured   unknown
```

例 1　各スロットの値の出力　　*(続き)*

```
c2              scsi-bus     connected    unconfigured   unknown
cpci_slot1      stpci/fhs    connected    configured     ok
cpci_slot2      unknown      empty        unconfigured   unknown
cpci_slot4      stpci/fhs    connected    configured     ok
cpci_slot5      stpci/fhs    connected    configured     ok
```

例 2　カードの交換

次のコマンドは、すべての DR 対応の接続点を一覧表示します。

```
example# cfgadm
```

```
Type            Receptacle   Occupant     Condition
c0              scsi-bus     connected    configured     unknown
c1              scsi-bus     connected    unconfigured   unknown
c2              scsi-bus     connected    unconfigured   unknown
cpci_slot1      stpci/fhs    connected    configured     ok
cpci_slot2      unknown      empty        unconfigured   unknown
cpci_slot4      stpci/fhs    connected    configured     ok
cpci_slot5      stpci/fhs    connected    configured     ok
```

次のコマンドは、カードを構成解除して電気的に切断します。

```
example# cfgadm -c disconnect cpci_slot4
```

次のコマンドを入力して変更を確認できます。

```
example# cfgadm cpci_slot4
```

```
Ap_Id                  Type         Receptacle   Occupant     Condition
cpci_slot4             unknown      disconnected unconfigured unknown
```

この時点で、カードを交換できるようになります。次のコマンドは、カードを電気的に接続して構成します。

```
example# cfgadm -c configure cpci_slot4
```

次のコマンドを入力すると、変更を検証できます。

```
example# cfgadm cpci_slot4
```

```
Ap_Id                  Type         Receptacle   Occupant     Condition
cpci_slot4             stpcipci/fhs connected    configured   ok
```

ファイル

`/usr/lib/cfgadm/pci.so.1`
PCI ホットプラグ操作のためのハードウェア固有のライブラリです。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

`cfgadm(1M)`, `cfgadm_shp(1M)`, `config_admin(3CFGADM)`, `libcfgadm(3LIB)`, `attributes(5)`

『Oracle Solaris の管理: 一般的なタスク』

名前

cfgadm_sata – cfgadm の SATA ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options]
     -c function ap_id...

/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_options]
     -x hardware_function ap_id...

/usr/sbin/cfgadm [-v] [-a] [-s listing_options]
     [-o hardware_options] [-l [ap_id | ap_type]...]

/usr/sbin/cfgadm [-v] [-o harware_options] -t ap_id...

/usr/sbin/cfgadm [-v] [-o hardware_options] -h [ap_id]...
```

機能説明

SATA ハードウェア固有のライブラリ `/usr/lib/cfgadm/sata.so.1` は、`cfgadm` コマンドによる SATA ホットプラグ操作のための機能を提供します。`cfgadm` は、接続点で実行されます。接続点とは、ハードウェアリソースを動的に再構成できるシステム上の場所です。接続点については、`cfgadm(1M)` を参照してください。

各 SATA コントローラおよびポートマルチプライヤのデバイスポートは、デバイスツリー上で接続点によって表されます。システム上で接続および構成されている SATA デバイスは、接続点の名前の拡張として表されます。次の説明では、「接続点」および「SATA ポート」という用語は同じ意味で使用されます。

接続点は `ap_id` を使用して指定されます。すべての SATA 接続点 `ap_id` は、次の形式の文字列で構成されます。

```
sataX/P[.M][::dsk/cXtYd0]
```

各情報の意味は次のとおりです。

X	SATA コントローラ番号です。
P	SATA コントローラのデバイスポート番号 (0 ～ 31) です。
M	ポートマルチプライヤのデバイスポート番号 (0 ～ 14)、ポートマルチプライヤのホストポート番号 (15) です。ポートマルチプライヤが SATA コントローラのデバイスポートに接続されている場合にのみ使用されます。
dev/cXtYd0	接続された SATA デバイスを識別します。
Y	ターゲット番号です。

一般的に、デバイスの識別子は、*/dev* 内のデバイスの対応する論理リンクから生成されます。SATA デバイスでは 1 つの LUN (LUN 0) しかサポートされないため、デバイス文字列の「d」コンポーネントは常に数値 0 (ゼロ) になります。

たとえば、SATA コントローラ 2 のデバイスポート 5 に接続されたポートマルチプライヤのデバイスポート 4 の論理 `ap_id` は次のとおりです。

```
sata2/5.4
```

この接続点に SATA ディスクまたは CD/DVD デバイスが接続されていて、デバイスが構成されている場合、*ap_id* は次のようになります。

```
sata2/5.4::dsk/c2t645d0
```

デバイスを識別する *cXtYd0* 文字列は、デバイスの接続点と 1 対 1 で対応します。

システム上の接続点の単純な一覧には、すべての SATA デバイスポートおよび接続されているデバイスが含まれます。例:

```
#cfgadm -l
Ap_Id                      Type        Receptacle   Occupant     Condition
  sata0/0::dev/c0t0d0      disk        connected    configured   ok
  sata0/1::dev/c0t1d0      disk        connected    configured   ok
  sata0/2::dev/c0t2d0      cd-dvd      connected    configured   ok
  sata0/3                  sata-port   empty        unconfigured ok
  sata1/0                  sata-port   disconnected unconfigured unknown
  sata1/1                  sata port   disconnected unconfigured unknown
  sata1/2                  sata port   empty        unconfigured ok
  sata1/3.15               sata-pmult  connected    configured   ok
  sata1/3.0::dev/c0t512d0 disk         connected    configured   ok
  sata1/3.1                sata-port   empty        unconfigured ok
  sata1/3.2                sata-port   empty        unconfigured ok
  sata1/3.3                sata-port   empty        unconfigured ok
  usb0/1                   unknown     empty        unconfigured ok
  usb0/2                   unknown     empty        unconfigured ok
```

接続点の一覧表示については、`cfgadm(1M)` を参照してください。

SATA ポートにおける接続点の受容体の状態は、次のような意味を持ちます。

empty
: SATA ポートが電源投入され、有効になっています。このポートでは、デバイスの存在が検出されませんでした。

disconnected
: SATA ポートが有効になっていない、SATA デバイスの存在は検出されたがデバイスとの通信が確立されてない、またはポートで障害が発生している、のいずれかです。

connected
: ポートで SATA デバイスが検出され、デバイスとの通信が確立されています。

占有装置 (SATA ポートに接続されたデバイス)の状態は、次のような意味を持ちます。

configured
: 接続されている SATA デバイスは構成されており、オペレーティングシステムが使用できる状態です。

unconfigured
: デバイスが接続されていないか、SATA ポートに接続されている SATA デバイスがまだ構成されていません。構成するには、コマンド「`cfgadm -c configure ap_id`」を実行します。

接続点 (SATA ポート) の状況は、次の意味を持ちます。

ok　SATA ポートが電源投入され、有効になっていて、使用できます。

failed　SATA ポートに障害が発生しました。システムにより無効になっているか電源が切断されている可能性があります。使用不可能であり、その状況は不明です。差し込まれているデバイスが原因である可能性があります。

unknown　SATA ポートが無効になっており、その状況は不明です。

「状態テーブル」は、接続点の受容体の状態、占有装置の状態、および接続点 (SATA ポート) の状況を組み合わせたものです。有効な状態は次のとおりです。

empty/unconfigured/ok	SATA ポートが有効になっており、アクティブです。デバイスの存在が検出されませんでした。
disconnected/unconfigured/ok	SATA ポートが有効になっていて、デバイスの存在が検出されましたが、デバイスとの通信は確立されませんでした。
disconnected/unconfigured/unknown	SATA ポートが無効になっており、その状況は不明です。
disconnected/unconfigured/failed	SATA ポートが無効になっており、使用不可です。システムで検出された障害が原因で、ポートはシステムで無効になりました。
connected/unconfigured/ok	SATA ポートが有効になっており、アクティブです。デバイスの存在が検出され、デバイスとの通信が確立されました。デバイスは OS で使用されるように構成されていません。
connected/configured/ok	デバイスは存在し、構成されており、OS が使用できる状態です。

オプション

`cfgadm` では、一覧表示 (`-l`) のほかに数種類の操作が定義されています。これらの操作には、テスト (`-t`)、構成状態の変更の開始 (`-c`)、ハードウェア固有の関数の実行 (`-x`)、および構成管理のヘルプメッセージの表示 (`-h`) が含まれます。

`-c` *function*

次の汎用の関数は、SATA ハードウェア固有のライブラリに定義されています。SATA ポート接続点では、次の構成状態の変更操作がサポートされています。

connect

SATA ポートを有効 (アクティブ) にして、接続されたデバイスとの通信を確立します。この操作により、必要に応じてポートの電源が投入されます。

disconnect
接続されたデバイスがまだ構成解除されていない場合は構成解除し、SATA ポートを無効化 (非アクティブ化) します。その後に「connect」コマンドを使用した場合、SATA ポート操作は有効になりますが、デバイスは「configured」状態にはなりません。

SATA ポートに接続された SATA デバイスでは、次の状態の変更操作がサポートされています。

configure　新しいデバイスがまだ構成されていない場合に、オペレーティングシステムが使用できるように構成します。このコマンドは、必要に応じて接続操作も実行することを意味しています。

unconfigure　SATA ポートに接続されたデバイスがまだ構成解除されていない場合は、構成解除します。

configure および unconfigure 操作は、ポートパルチプライヤが接続された接続点では使用できません。ポートパルチプライヤは、システムで自動的に構成および構成解除されません。ただし、configure および unconfigure 操作は、ポートパルチプライヤのデバイスポートに接続されたすべての SATA デバイスに適用されます。

-f
サポートされていません。

-h *ap_id*
ヘルプオプションと任意の SATA 接続点を組み合わせて使用すると、SATA 固有のヘルプを取得できます。

-l [-v]
-l オプションは、cfgadm(1M) で説明されているように機能します。-v オプションと組み合わせて使用すると、「Information」フィールドに次の SATA 固有の情報が表示されます。

- Mfg: 製造元文字列
- Product: 製品文字列
- No: 製品シリアル番号

-o*hardware_options*
現在定義されているハードウェア固有のオプションはありません。

-s*listing_options*
select サブオプションを使用すると、クラス SATA の接続点を一覧表示できます。cfgadm(1M) を参照してください。

-t *ap_id*
SATA コントローラでサポートされている場合、SATA ポートのセルフテストを実行します。セルフテスト操作が SATA コントローラでサポートされていない場合は、エラーメッセージが表示されます。

-x*hardware_function*
ハードウェア固有の関数を実行します。

前述のとおりに、SATA ポートまたは SATA コントローラの次のコマンドの一部では、接続されている SATA デバイスが影響を受ける可能性があります。前述のとおりに、`ap_id` は SATA ポートまたは SATA コントローラ全体を指します。操作にデバイスの構成解除が含まれていても構成解除できない場合 (つまり、デバイスにマウント済みのファイルシステムが含まれる場合)、エラーメッセージが表示され、操作は実行されません。SATA コントローラで指定された操作がサポートされていない場合も、エラーメッセージが表示されます。

sata_reset_device ap_id
`ap_id` SATA ポートに接続された SATA デバイスをリセットします。SATA ポートの状態は変更されません。

sata_reset_port ap_id
`ap_id` で指定された SATA ポートをリセットします。ポートに SATA デバイスが接続されている場合は、そのデバイスもリセットされます。この操作は、ポートマルチプライヤが接続されているポートでも実行される場合があります。ポートマルチプライヤが SATA コントローラポートに接続されている場合、ポートマルチプライヤに接続された SATA デバイスがリセットされない可能性があります。

sata_reset_all ap_id
`ap_id` のコントローラ番号部分で指定された SATA コントローラおよび接続されたすべてのデバイスをリセットし、接続されたすべてのデバイス (ポートマルチプライヤやポートマルチプライヤのデバイスポートに接続されたデバイスなど) を再列挙します。

この操作により、操作前に接続されたすべてのデバイスが接続解除されます。新しく列挙されたデバイスは構成解除されたままです。

sata_port_deactivate ap_id
ほかのすべてに失敗した場合に、ポートを強制的に無効にします。これは、緊急時の手順として使用します。注意して使用してください。

sata_port_activate ap_id
ポートを強制的に有効にします。これは、エラーから回復するために無効にされたポート上で緊急状況時に使用されます。

sata_port_self_test ap_id
SATA コントローラでセルフテスト操作を実行します。この操作により、すべてのデバイスが構成解除され、SATA コントローラがリセットされます。

-v
詳細モードで実行します。

次の「遷移」テーブルは、`-c` 操作およびホットプラグ処理の結果生じた状態遷移を報告します。

```
current state     operation       possible new state
-------------     ---------       ------------------
empty/
unconfigured/ok   device plug-in  connected/unconfigured/ok, or
                                  disconnected/unconfigured/ok, or
                                  disconnected/unconfigured/failed

empty/
unconfigured/ok   -c unconfigure  error message, no state change

empty/
unconfigured/ok   -c configure    error message, no state change

empty/
unconfigured/ok   -c connect      error message, no state change

empty/
unconfigured/ok   -c disconnect   disconnected/unconfigured/unknown, or
                                  disconnected/unconfigured/failed

disconnected/
unconfigured/ok   device unplug   no state change

disconnected/
unconfigured/ok   -c unconfigure  error message, no state change

disconnected/
unconfigured/ok   -c configure    error message, no state change

disconnected/
unconfigured/ok   -c connect      error message, no state change

disconnected/
unconfigured/ok   -c disconnect   error message, no state change

disconnected/
unconfigured/
unknown
(no disk plugged) -c configure    error message, state change to
                                  empty/unconfigured/ok, or
                                  disconnected/unconfigured/failed

disconnected/
unconfigured/
unknown           -c configure    state change to
(disk plugged)                    connected/configured/ok or,
                                  connected/unconfigured/ok, or
                                  disconnected/unconfigured/failed and
```

```
                                    possible error message

disconnected/
unconfigured/
unknown           -c connect      empty/unconfigured/ok, or
                                  connected/unconfigured/ok, or
                                  disconnected/unconfigured/ok, or
                                  disconnected/unconfigured/unknown, or
                                  disconnected/unconfigured/failed

disconnected/
unconfigured/
unknown           -c disconnect   error message, no state change

disconnected/
unconfigured/
failed            any command     error message, no state change
                  other than
                  -x commands

connected/
unconfigured/ok   disk unplug     error message and state:
                                  empty/unconfigured/ok, or
                                  disconnected/unconfigured/failed

connected/
unconfigured/ok   -c configure    connected/unconfigured/ok, or
                                  connected/configured/ok, or
                                  disconnected/unconfigured/ok, or
                                  disconnected/unconfigured/failed

connected/
unconfigured/ok   -c unconfigure  error message, no state change

connected/
unconfigured/ok   -c connect      error message, no state change

connected/
unconfigured/ok   -c disconnect   disconnected/unconfigured/unknown, or
                                  disconnected/unconfigured/failed

connected/
configured/ok     disk unplug     error message and state:
                                  empty/unconfigured/ok, or
                                  disconnected/unconfigured/failed

connected/
configured/ok     -c configure    error message, no state change
```

```
connected/
configured/ok     -c unconfigure  error message, if device cannot be
                                  unconfigured, no state change, or
                                  connected/unconfigured/ok, or
                                  disconnected/unconfigured/ok, or
                                  disconnected/unconfigured/failed

connected/
configured/ok     -c connect      error message, no state change

connected/
configured/ok     -c disconnect   error message, if device cannot be
                                  unconfigured, no state change, or
                                  disconnected/unconfigured/unknown, or
                                  disconnected/unconfigured/failed
```

使用例

例 1 ディスクの構成

次のコマンドは、SATA コントローラ 0、ポート 0 に接続されているディスクを構成します。

```
example# cfgadm -c configure sata0/0
```

このコマンドは、SATA ポートにデバイスが接続されている場合にのみ発行してください。

例 2 ディスクの構成解除

次のコマンドは、SATA コントローラ 0、ポート 3 に接続されたディスクを構成解除します。

```
example# cfgadm -c unconfigure sata0/3
```

デバイスを識別する文字列は、接続点の受容体の状態が「connected」、占有装置の状態が「configured」のときに表示されます。

例 3 マウントされたファイルシステムがディスクの構成解除中に検出された場合

次のコマンドは、マウントされたファイルシステムがディスクの構成解除中に検出された場合を示しています。

```
example# cfgadm -c unconfigure sata1/5
```

システムは次のように応答します。

```
cfgadm: Component system is busy, try again: failed to offline:
/devices/pci@0,0/pci8086,244e@1e/pci1095,3124@1/sd@5,0
     Resource              Information
------------------  --------------------------
/dev/dsk/c1t5d0s0   mounted filesystem "/mnt"
```

ファイル

/usr/lib/cfgadm/sata.so.1　汎用的な SATA ホットプラグ操作のためのハードウェア固有ライブラリ

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

cfgadm(1M), config_admin(3CFGADM), libcfgadm(3LIB), attributes(5)

注意事項

緊急時の「sata_port_deactivate」操作は、ルート (/)、/usr、swap、/var などの重要なパーティションを含むディスクが接続されたポートではサポートされていません。このようなポートでは、無効化操作を実行しないでください。使用法を誤ると、システムがハングアップし、リブートが必要になる場合があります。

ホットプラグ操作は、すべての SATA コントローラでサポートされているわけではありません。

SATA コネクタがホットプラグ対応のタイプであり、SATA コントローラでホットプラグがサポートされている場合は、いつでも SATA デバイスのホットプラグ操作が可能です。システムはイベントを検出して、デバイスとの通信を確立します。デバイスは、明示的に「`cfgadm -c configure ap_id`」コマンドで構成する必要があります。

SATA コネクタがホットプラグ対応のタイプであり、SATA コントローラでホットプラグがサポートされている場合、構成解除せずにデバイスを取り外すと、システムがハングアップしたり、データが損失したりする可能性があります。デバイスが構成解除されたが、受容体の状態が disconnected 状態ではない場合は、SATA ポートからデバイスを取り外すと、エラーメッセージが表示されます。

警告

一部の SATA デバイスのコネクタは、SATA ホットプラグの仕様に準拠していません。このようなデバイスでホットプラグ操作を実行すると、SATA コントローラまたは SATA デバイス (あるいはその両方) に障害を与える可能性があります。

名前

cfgadm_sbd – システムボード管理用の cfgadm コマンド

形式

```
cfgadm  -l [-a] [-o parsable]  ap_id...

cfgadm  -c  function [-f] [-y | -n]
     [-o unassign | nopoweroff] [-v]  ap_id...

cfgadm  -t [-v]  ap_id...

cfgadm  -x  [-f] [-v] function  ap_id...
```

機能説明

cfgadm_sbd プラグインは、クラス sbd のシステムボードを接続、構成、構成解除、および切断するための動的な再構成機能を提供します。このプラグインを使用すると、システムをリブートすることなく、システムボードと実行中のシステムを接続または切断することもできます。

cfgadm コマンドは /usr/sbin に存在します。cfgadm(1M) を参照してください。cfgadm_sbd プラグインは /usr/platform/sun4u/lib/cfgadm に存在します。

各ボードスロットは、デバイスツリーでは単一の接続点として表示されます。各コンポーネントは、動的接続点として表示されます。-a オプションを使用すると、各コンポーネントのタイプ、状態、条件、および各ボードスロットの状態と条件を表示できます。

cfgadm オプションは、プラットフォームに応じて動作が異なります。また、接続点の形式もプラットフォームに応じて異なります。詳細は、「プラットフォームにおける注意事項」の節を参照してください。

コンポーネントの条件

コンポーネントの条件の名前と説明を次に示します。

failed
: コンポーネントでテストが失敗しました。

ok
: コンポーネントは稼働しています。

unknown
: コンポーネントはテストされませんでした。

コンポーネントの状態

コンポーネントの受容体の状態の名前と説明を次に示します。

connected
: コンポーネントはボードスロットに接続されています。

コンポーネントの占有装置の状態の名前と説明を次に示します。

configured
: コンポーネントは、Solaris オペレーティング環境で使用できる状態です。

unconfigured
: コンポーネントは、Solaris オペレーティング環境で使用できる状態ではありません。

ボードの条件

ボードの条件の名前と説明を次に示します。

failed
ボードでテストが失敗しました。

ok
ボードは稼働しています。

unknown
ボードはテストされませんでした。

unusable
ボードスロットは使用できません。

ボードの状態

ボードを挿入すると、受容体の状態は empty から disconnected に変わります。ボードを取り外すと、受容体の状態は disconnected から empty に変わります。

注意: connected 状態のボード、または電源が投入されていて disconnected 状態のボードを取り外すと、オペレーティングシステムがクラッシュし、システムに修復不可能な障害を与える場合があります。

ボードの受容体の状態の名前と説明を次に示します。

connected
ボードは電源が投入されており、システムバスに接続されています。ボードが connected 状態になった場合のみ、ボード上のコンポーネントが表示されます。

disconnected
ボードはシステムバスから切断されています。ボードは、電源を切断しなくても disconnected 状態にできます。ただし、スロットから取り外す前に、ボードの電源を切断し、disconnected 状態にする必要があります。

empty
ボードが存在しません。

切断されたボードの占有装置の状態は常に unconfigured です。ボードの占有装置の状態の名前と説明を次の表に示します。

configured
ボード上の少なくとも 1 つのコンポーネントが構成されています。

unconfigured
ボード上のすべてのコンポーネントが構成解除されています。

ダイナミックシステムドメイン

ダイナミックシステムドメイン (Dynamic System Domain、DSD。このマニュアルではドメインと呼びます) を基盤とするプラットフォームでは、シャーシ内のスロットが、電気的に絶縁されたハードウェアパーティション (つまり、DSD) に分割されます。DSD を基盤としないプラットフォームでは、すべてのスロットがシステムに恒久的に割り当てられます。

スロットは空けておくことも使用することもできます。また、任意の数のドメインに割り当てることも、任意の数のドメインで使用できるようにすることもできます。所定のドメインで使用できるスロットの数は、システムコントローラで維持される使用可能なコンポーネントのリスト (Available Component List、ACL) で制御されます。ACL は、Solaris オペレーティング環境によって提供されるアクセス制御リストではありません。

スロットがドメインの ACL に含まれ、スロットが別のドメインに割り当てられていない場合にのみ、ドメインからスロットを認識できます。割り当てられていないスロットは、そのスロットが ACL に含まれているすべてのドメインから認識できます。スロットが特定のドメインに割り当てられると、そのスロットはほかのドメインから認識できなくなります。

あるドメインで認識可能であるが、未割り当てのスロットは、まずそのドメインに割り当てをしてから、ほかの状態変更コマンドを適用する必要があります。割り当ては、`-x assign` を使用して明示的に行うか、接続の一環として暗黙的に行うことができます。スロットを別のドメインで使用できるようにするには、特定のドメインから割り当て解除する必要があります。割り当て解除は常に明示的であり、`-x unassign` を使用して直接行うか、または `-o unassign` を使用して切断オプションとして行います。

状態変更機能

ボードスロットまたはボード上のコンポーネントの状態を変更する機能を、任意の接続点に対し同時に発行できます。一度に許可される状態変更操作は 1 つのみです。状態変更情報の Busy フィールドの `Y` は、操作が進行中であることを示します。

次の一覧は、状態を変更する機能を示しています。

- configure
- unconfigure
- connect
- disconnect

可用性変更機能

ボードの可用性を変更するコマンドを、任意の接続点に対し同時に発行できます。一度に許可される可用性変更操作は 1 つのみです。これらの機能により、`cfgadm -l` の出力の情報文字列も変更されます。Busy フィールドの `Y` は、操作が進行中であることを示します。

次の一覧は、可用性を変更する機能を示しています。

- `assign`
- `unassign`

条件変更機能

ボードスロットまたはボード上のコンポーネントの条件を変更する機能を、任意の接続点に対し同時に発行できます。一度に許可される条件変更操作は 1 つのみです。これらの機能により、`cfgadm -l` の出力の情報文字列も変更されます。Busy フィールドの `Y` は、操作が進行中であることを示します。

次の一覧は、条件を変更する機能を示しています。

- poweron
- poweroff
- test

構成解除プロセス

この節では、構成解除プロセスについて説明し、永続メモリーを移動するプロセス中のさまざまな段階におけるソースボードおよびターゲットボードの状態を示します。

次のコード例では、ボード 0 上の永続メモリーをドメイン内の別のボードに移動する必要があります。したがって、ボード 0 がソース、ボード 1 がターゲットです。

ボードがビジーとマークされている間は、そのボードでステータス変更操作を開始することはできません。簡潔にするため、CPU 情報はコード例から削除されています。

プロセスは次のコマンドで開始されます。

```
# cfgadm -c unconfigure -y SB0::memory &
```

最初に、ボード 0 上の永続メモリーと同じアドレス範囲にあるボード 1 上のメモリーを削除する必要があります。この段階では、ソースボード、ターゲットボード、およびメモリーの接続点がビジーとマークされています。次のコマンドでステータスを表示できます。

```
# cfgadm -a -s cols=ap_id:type:r_state:o_state:busy SB0 SB1

Ap_Id          Type      Receptacle     Occupant       Busy
SB0            CPU       connected      configured     y
SB0::memory    memory    connected      configured     y
SB1            CPU       connected      configured     y
SB1::memory    memory    connected      configured     y
```

ボード 1 上のメモリーを削除すると、unconfigured とマークされます。次の例に示すように、ボード 0 上のメモリーは configured のままですが、引き続きビジーとマークされています。

```
Ap_Id          Type      Receptacle     Occupant       Busy
SB0            CPU       connected      configured     y
SB0::memory    memory    connected      configured     y
SB1            CPU       connected      configured     y
SB1::memory    memory    connected      unconfigured   n
```

次に、メモリーがボード 0 からボード 1 にコピーされます。コピーされると、メモリーの占有装置の状態が切り替わります。ボード 0 上のメモリーは unconfigured になり、ボード 1 上のメモリーは configured になります。次の例に示すように、プロセスのこの時点では、ボード 0 だけがビジーのままです。

```
Ap_Id          Type      Receptacle     Occupant       Busy
SB0            CPU       connected      configured     y
SB0::memory    memory    connected      unconfigured   n
SB1            CPU       connected      configured     n
SB1::memory    memory    connected      configured     n
```

次の例に示すように、プロセス全体が完了すると、ボード 0 上のメモリーは unconfigured のままとなり、接続点はビジーではありません。

```
Ap_Id          Type      Receptacle     Occupant       Busy
SB0            CPU       connected      configured     n
SB0::memory    memory    connected      unconfigured   n
SB1            CPU       connected      configured     n
SB1::memory    memory    connected      configured     n
```

永続メモリーが移動され、ボード 0 上のメモリーが構成解除されました。この時点で、いずれのボードでも新しい状態変更操作を開始できます。

プラットフォーム固有のオプション

システムボードプラグインによって解釈されるオプションの後に配置する、プラットフォーム固有のオプションを指定できます。すべてのプラットフォーム固有のオプションの前に、`platform` キーワードを指定する必要があります。次の例は、プラットフォーム固有のオプションを指定したコマンドの一般的な形式を示しています。

command `-o` *sbd_options*`,platform=`*platform_options*

オプション

このマニュアルページでは、`cfgadm` コマンドの `-v`、`-a`、`-s`、または `-h` オプションについては説明していません。これらのオプションの説明については、`cfgadm`(1M) を参照してください。`cfgadm_sbd` プラグインでは、次のオプションがサポートされています。

`-c` *function*
: 状態変更機能を実行します。次の機能を使用できます。

unconfigure
: 占有装置の状態を unconfigured に変更します。この機能は、システムボードスロット、およびシステムボード上のすべてのコンポーネントに適用されます。

`unconfigure` 機能により、CPU が CPU リストから削除され、物理メモリーがシステムメモリープールから削除されます。いずれかのデバイスがまだ使用中の場合、`cfgadm` コマンドは失敗し、ユーザーにエラーが報告されます。デバイスのビジー状態が終わりしだい、コマンドを再試行できます。CPU が使用中の場合は、オフラインであることを確認してから操作を進める必要があります。`pbind`(1M)、`psradm`(1M)、および `psrinfo`(1M) を参照してください。

unconfigure 機能は、物理メモリーを別のシステムボードに移動してから、構成解除するボードからメモリーを削除します。移動するメモリーの種類によっては、別のボード上に十分なメモリーが見つからない場合に、または適切な物理メモリー範囲が見つからない場合に、コマンドが失敗します。

永続メモリーの場合、メモリーが移動され、メモリーコントローラが再プログラムされている間は、オペレーティングシステムを停止 (つまり、休止) する必要があります。オペレーティングシステムを停止する必要がある場合、操作の続行を確認するプロンプトが表示されます。「はい」または「いいえ」と答えるには、それぞれ -y または -n オプションを使用します。

メモリーの量およびシステム負荷によっては、メモリーの移動に数分かかる場合があります。メモリーの接続点に対して status コマンドを発行すると、操作の進行状況を監視できます。cfgadm コマンドを停止して、メモリー操作を中断することもできます。削除されたメモリーは、システムメモリープールに戻されます。

disconnect

受容体の状態を disconnected に変更します。この機能は、システムボードスロットのみに適用されます。

占有装置の状態が configured になっている場合は、占有装置の構成解除を試みます。次に、システムボードの電源を切断します。この時点で、ボードをスロットから取り外すことができます。

ダイナミックシステムドメインをサポートするプラットフォームでは、この機能は、ボードを割り当てられた状態のままにします。

-o nopoweroff を指定した場合、disconnect 機能は、ボードを電源が投入された状態のままにします。-o unassign を指定した場合、disconnect 機能は、ドメインからボードを割り当て解除します。

ドメインからボードを割り当て解除した場合、そのボードを別のドメインに割り当てることができます。ただし、別のドメインに割り当てた場合、割り当て解除されたドメインはそのボードを使用できなくなります。

configure

占有装置の状態を configured に変更します。この機能は、システムボードスロット、およびシステムボード上のすべてのコンポーネントに適用されます。

受容体の状態が disconnected になっている場合は、configure 機能は受容体への接続を試みます。その後、connect 機能によって作成されたデバイスツリーを順に調べていき、必要に応じてデバイスを接続します。この機能を実行すると、すでに構成済みのコンポーネントを除き、ボード上のすべてのコンポーネントが構成されます。

CPU の場合、configure 機能により、CPU が CPU リストに追加されます。メモリーの場合、configure 機能は、メモリーが初期化されているのを確認してから、メモリーをシステムメモリープールに追加します。configure 機能が正常に完了すると、CPU とメモリーは使用できる状態になっています。

I/O デバイスの場合、デバイスを使用できるようにするには、mount コマンドと ifconfig コマンドを使用する必要があります。ifconfig(1M) および mount(1M) を参照してください。

connect
受容体の状態を connected に変更します。この機能は、システムボードスロットのみに適用されます。

ボードスロットがドメインに割り当てられていない場合、connect 機能は、ドメインへのスロットの割り当てを試みます。次に、ボードの電源を投入し、テストして、ボードをシステムバスに電気的に接続し、コンポーネントを調べます。

connect 機能が正常に完了したら、-a オプションを使用して、ボード上のコンポーネントのステータスを表示できます。connect 機能は、すべてのコンポーネントを unconfigured 状態のままにします。

割り当て手順は、ダイナミックシステムドメインをサポートするプラットフォームのみに適用されます。

-f
ソフトウェアの状態の変更に関する制約を無効化します。

-f によって、ハードウェアおよびオペレーティングシステムの基本的な安全および可用性に関する制約が無効化されることはありません。

-l
cfgadm(1M) で指定されている -s、-v、および -a の各オプションによって制御される形式で指定された接続点の状態や条件を一覧表示します。後述するように、cfgadm_sbd プラグインにより、info フィールドに固有の情報が示されます。この情報の形式は、-o parsable オプションによって変更できます。

構文解析可能な info フィールドは、次で構成されます。

cpu
cpu タイプでは、次の情報が表示されます。

cpuid=#[,#…]
ここで、# は数値であり、CPU の ID を表します。複数の # が存在する場合、この CPU には、複数のアクティブな仮想プロセッサがあります。

speed=#
ここで、# は数値であり、CPU の速度を MHz で表します。

`ecache=#`
ここで、#は数値であり、外部キャッシュのサイズをMバイトで表します。CPUに複数のアクティブな仮想プロセッサがある場合、外部キャッシュは仮想プロセッサ間で共有されるか、仮想プロセッサ間で分割されます。

memory
`memory`タイプでは、必要に応じて次の情報が表示されます。

address=#
ここで、#は数値であり、ベース物理アドレスを表します。

size=#
ここで、#は数値であり、メモリーのサイズをKバイトで表します。

permanent=#
ここで、#は数値であり、永続メモリーのサイズをKバイトで表します。

unconfigurable
メモリーを構成解除できないようにするオペレーティングシステム設定です。

inter-board-interleave
ボードが、ほかのボードとのインタリーブに参加しています。

source=*ap_id*
ソースの接続点を表します。

target=*ap_id*
ターゲットの接続点を表します。

deleted=#
ここで、#は数値であり、すでに削除されたメモリーの量をKバイトで表します。

remaining=#
ここで、#は数値であり、削除されるメモリーの量をKバイトで表します。

io
`io`タイプでは、次の情報が表示されます。

device=*path*
I/Oコンポーネントの物理パスを表します。

referenced
I/Oコンポーネントは参照されています。

board
`board`タイプでは、次のブール型の名前が表示されます。これらが存在しない場合は、反対の意味になります。

assigned
ボードがドメインに割り当てられています。

powered-on
ボードに電源が投入されています。

`-o parsable` オプションを指定しない場合、`info` フィールドには、同じ項目がより読みやすい形式で表示されます。

`-o` parsable
ブール型の名前、または、スペース文字で区切られた一連の名前=値のペアとして `info` フィールドに情報を返します。

`-o parsable` オプションは、`-s` オプションと組み合わせて使用できます。`-s` オプションについては、`cfgadm(1M)` のマニュアルページを参照してください。

`-t`
ボードをテストします。

ボードは、適切なレベルのテストをパスしなければ、接続できません。

このオプションを使用すると、適切なテストレベルをすでにパスしていても、必ずボードのテストが試みられます。`-c connect` 状態変更機能が発行されたときにもテストは実行されます。この場合で、ボードに適切なテストレベルがすでに示されているときは、テスト手順が省略されます。したがって、`-t` オプションを使用すると、ボードがテストされるように明示的に要求できます。

`-x` *function*
sbd クラス機能を実行します。次の機能を使用できます。

assign
ドメインにボードを割り当てます。

受容体の状態は disconnected または empty である必要があります。また、ボードは、ドメインの使用可能なコンポーネントのリストに含まれている必要があります。「ダイナミックシステムドメイン」を参照してください。

unassign
ドメインからボードを割り当て解除します。

受容体の状態は disconnected または empty である必要があります。また、ボードは、ドメインの使用可能なコンポーネントのリストに含まれている必要があります。「ダイナミックシステムドメイン」を参照してください。

poweron
システムボードに電源を投入します。

受容体の状態は disconnected である必要があります。

poweroff
システムボードの電源を切断します。

受容体の状態は disconnected である必要があります。

オペランド

次のオペランドがサポートされています。

Receptacle *ap_id*
Sun Fire 15K などの Sun Fire ハイエンドシステムの場合は、受容体の接続点 ID は、`SB`*X* または `IO`*X* の形式をとります。ここで、*X* はスロット番号です。

正確な形式はプラットフォームによって異なり、通常は、マシン上の物理ラベルに対応します。「注意事項」の節のプラットフォーム固有の情報を参照してください。

Component *ap_id*
コンポーネントの接続点 ID は、*component_typeX* の形式をとります。ここで、*component_type* は、「コンポーネントタイプ」で説明されているいずれかのコンポーネントタイプであり、*X* はコンポーネント番号です。コンポーネント番号は、ボードを基準としたユニット番号です。

上記の規則は、メモリーコンポーネントには適用されません。メモリーの接続点に対する DR 操作は、いずれもシステムボード上のすべてのメモリーに影響します。

使用例

次の例は、Sun Fire 15K システムでのユーザー入力とシステム出力を示しています。ユーザー入力 (具体的には、接続点に対する参照) とシステム出力は、6800 のような Sun Fire ミッドレンジシステムなどのほかの Sun Fire システムでは異なる場合があります。Sun Fire ハイエンドモデル以外での `cfgadm_sbd` プラグインの使用に関する具体的な情報については、プラットフォームにおける注意事項を参照してください。

例 1 すべてのシステムボードの一覧表示

```
# cfgadm -a -s "select=class(sbd)"

Ap_Id          Type      Receptacle     Occupant       Condition
SB0            CPU       connected      configured     ok
SB0::cpu0      cpu       connected      configured     ok
SB0::memory    memory    connected      configured     ok
IO1            HPCI      connected      configured     ok
IO1::pci0      io        connected      configured     ok
IO1::pci1      io        connected      configured     ok
SB2            CPU       disconnected   unconfigured   failed
SB3            CPU       disconnected   unconfigured   unusable
SB4            unknown   empty          unconfigured   unknown
```

この例は、次のそれぞれの状況を示しています。

- スロット 2 のボードでテストが失敗しました。

例 1 すべてのシステムボードの一覧表示 *(続き)*

- スロット 3 は使用できないため、そのスロットにボードをホットプラグすることはできません。

例 2 システムボード上のすべての CPU の一覧表示

```
# cfgadm -a -s "select=class(sbd):type(cpu)"

Ap_Id          Type       Receptacle     Occupant       Condition
SB0::cpu0      cpu        connected      configured     ok
SB0::cpu1      cpu        connected      configured     ok
SB0::cpu2      cpu        connected      configured     ok
SB0::cpu3      cpu        connected      configured     ok
```

例 3 CPU 情報フィールドの表示

```
# cfgadm -l -s noheadings,cols=info SB0::cpu0

cpuid 16, speed 400 MHz, ecache 8 Mbytes
```

例 4 構文解析可能な形式での CPU 情報フィールドの表示

```
# cfgadm -l -s noheadings,cols=info -o parsable SB0::cpu0

cpuid=16 speed=400 ecache=8
```

例 5 I/O ボード上のデバイスの表示

```
# cfgadm -a -s noheadings,cols=ap_id:info -o parsable IO1

IO1       powered-on assigned
IO1::pci0 device=/devices/saf@0/pci@0,2000 referenced
IO1::pci1 device=/devices/saf@0/pci@1,2000 referenced
```

例 6 構成解除操作の監視

次の例では、メモリーサイズが K バイトで表示されています。

```
# cfgadm -c unconfigure -y SB0::memory &
# cfgadm -l -s noheadings,cols=info -o parsable SB0::memory SB1::memory

address=0x0 size=2097152 permanent=752592 target=SB1::memory
     deleted=1273680 remaining=823472
address=0x1000000 size=2097152 source=SB0::memory
```

例 7 ドメインへのスロットの割り当て

```
# cfgadm -x assign SB2
```

例 8 ドメインからのスロットの割り当て解除

```
# cfgadm -x unassign SB3
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library/platform
安定性	下記を参照。

インタフェースの安定性は開発中です。出力の安定性は不安定です。

関連項目

cfgadm(1M), devfsadm(1M), ifconfig(1M), mount(1M), pbind(1M), psradm(1M), psrinfo(1M), config_admin(3CFGADM), attributes(5)

注意事項

この節では、メモリーの削除操作の進行状況を監視する方法について説明しています。また、プラットフォーム固有の情報も示しています。

メモリーの削除の監視

次のシェルスクリプトを使用すると、メモリーの削除操作の進行状況を監視できます。

```
# cfgadm -c unconfigure -y SB0::memory &
# watch_memdel SB0

#!/bin/sh
# This is the watch_memdel script.

if [ -z "$1" ]; then
        printf "usage:  %s board_id\n" ‘basename $0‘
        exit 1
fi

board_id=$1

cfgadm_info=’cfgadm -s noheadings,cols=info -o parsable’

eval ‘$cfgadm_info $board_id::memory‘

if [ -z "$remaining" ]; then
        echo no memory delete in progress involving $board_id
        exit 0
fi

echo deleting target $target

while true
do
```

```
        eval `$cfgadm_info $board_id::memory`

        if [ -n "$remaining" -a "$remaining" -ne 0 ]
        then
                echo $deleted KBytes deleted, $remaining KBytes remaining
                remaining=
        else
                echo memory delete is done
                exit 0
        fi
        sleep 1
done
exit 0
```

Sun Enterprise 10000 プラットフォームにおける注意事項

Sun Enterprise 10000 システム上の接続点を参照するには、次の構文を使用します。

board::*component*

ここで、*board* はシステムボードを指し、*component* は個々のコンポーネントを指します。システムボードの範囲は、SB0 (ゼロ) から SB15 までです。最大で 16 枚のシステムボードを使用できます。

Sun Enterprise 10000 ドメインを実行している DR 3.0 モデルでは、cfgadm_sbd プラグインによって提供される機能の一部がサポートされます。サポートされる唯一の操作は、ドメイン内の接続点のステータスを表示することです。これは、-l オプションと、これに関連付けられているすべてのオプションに相当します。

ドメインからほかの操作を実行しようとすると、操作がサポートされていないことを示すエラーが表示されます。システムボードを追加または削除するすべての操作は、システムサービスプロセッサから開始する必要があります。

Sun Fire ハイエンドシステムプラットフォームにおける注意事項

Sun Fire ハイエンドシステム上の接続点を参照するには、次の構文を使用します。

board::*component*

ここで、*board* はシステムボードまたは I/O ボードを指し、*component* は個々のコンポーネントを指します。

システムの構成に応じて、システムボードの範囲は SB0 (ゼロ) から SB17 までとなり、I/O ボードの範囲は IO0 (IO ゼロ) から IO17 までとなります (最大で 18 枚のシステムボードおよび I/O ボードを使用できます)。

Sun Fire ハイエンドシステムプラットフォームでは、-t オプションおよび -x オプションの動作が異なります。次の一覧で、これらの動作について説明します。

-t
: システムコントローラが、CPU を使用し、LPOST と hpost コマンドを順に実行して、システムボードをテストします。I/O ボードをテストするために、ドライバが -t オプションに応答してテストを開始すると、テストがユーザーの操作なし

で自動的に実行されます。ドライバは、CPU および一続きの物理的に連続したメモリーを構成解除します。その後、システムコントローラにコマンドを送信し、ボードをテストします。システムコントローラが、トランザクション/エラーケージの内側から CPU とメモリーを使用して、I/O ボードをテストします。I/O ボードのテストに使用できるのは、(MCPU ボードではなく) システムボードの CPU だけです。

`-x assign | unassign`
Sun Fire ハイエンドシステム管理モデルでは、各ドメインの使用可能なコンポーネントのリストを使用して、プラットフォーム管理者がプラットフォームハードウェアを制御します。この情報は、システムコントローラで保持されます。プラットフォーム管理者だけが、ドメインの使用可能なコンポーネントのリストを変更できます。

ドメイン管理者は、ボードがそのドメインの使用可能なコンポーネントのリストに含まれている場合にのみ、そのボードの割り当てまたは割り当て解除を行うことができます。プラットフォーム管理者にはこの制限はなく、ドメインの使用可能なコンポーネントのリストに含まれていない場合でも、ボードの割り当てまたは割り当て解除を行うことができます。

Sun Fire 15K のコンポーネントタイプ

コンポーネントタイプの名前と説明を次に示します。

`cpu`
CPU

`io`
I/O デバイス

`memory`
メモリー

注: メモリーコンポーネントに対する操作は、ボード上のすべてのメモリーコンポーネントに影響します。

Sun Fire ミッドレンジシステムプラットフォームにおける注意事項

6800、4810、4800、3800 システムなどの Sun Fire ミッドレンジサーバーと Sun Fire ハイエンドシステムでは、接続点への参照がわずかに異なります。Sun Fire 15K 以外の Sun Fire システム上の接続点を参照するには、次の構文を使用します。

`N#.`*board*`::`*component*

ここで、**`N#`** はノードを指し、*board* はシステムボードまたは I/O ボードを指し、*component* は個々のコンポーネントを指します。

システムの構成に応じて、システムボードの範囲は `SB0` から `SB5` までとなり、I/O ボードの範囲は `IB6` から `IB9` までとなります (最大で 6 枚のシステムボードと 4 枚の I/O ボードを使用できます)。

Sun Fire ミッドレンジシステムのコンポーネントタイプ

コンポーネントタイプの名前と説明を次に示します。

`cpu`
: CPU

`pci`
: I/O デバイス

`memory`
: メモリー

注: メモリーコンポーネントに対する操作は、ボード上のすべてのメモリーコンポーネントに影響します。

名前

cfgadm_scsi – cfgadm の SCSI ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_option]
     -c function ap_id...

/usr/sbin/cfgadm [-f] [-y | -n] [-v] [-o hardware_option]
     -x hardware_function ap_id...

/usr/sbin/cfgadm [-v] [-a] [-s listing_option] [-o hardware_option]
     [-l [ap_id | ap_type ... ]]

/usr/sbin/cfgadm [-v] [-o hardware_option] -t ap_id...

/usr/sbin/cfgadm [-v] [-o hardware_option] -h [ap_id]...
```

機能説明

SCSI ハードウェア固有ライブラリ `/usr/lib/cfgadm/scsi.so.1` は、cfgadm(1M) コマンドによる SCSI ホットプラグ操作のための機能を提供します。`cfgadm` は、接続点で実行されます。接続点とは、ハードウェアリソースを動的に再構成できるシステム上の場所です。接続点については、`cfgadm`(1M) を参照してください。

SCSI ホットプラグ操作では、各 SCSI コントローラは、デバイスツリー上で接続点によって表されます。また、各 SCSI デバイスは、動的接続点によって表されます。接続点には、`ap_id` を通じて名前が付けられます。論理 ap_id と物理 ap_id の 2 種類の `ap_id` が定義されます。物理 `ap_id` が物理パス名に基づいているのに対し、論理 `ap_id` は、より短く、ユーザーにわかりやすい名前です。SCSI コントローラの場合、論理 `ap_id` は、通常は対応するディスクコントローラ番号です。たとえば、一般的な論理 `ap_id` は `c0` となります。

SCSI デバイスには、コントローラの `ap_id` を基準として名前が付けられます。したがって、ディスクデバイスがコントローラ `c0` に接続されている場合、その `ap_id` は次のようになります。

```
c0::dsk/c0t0d0
```

ここで、`dsk/c0t0d0` は、特定のデバイスを識別します。一般的に、デバイスの識別子は、`/dev` 内のデバイスの対応する論理リンクから生成されます。たとえば、SCSI テープドライブの論理 `ap_id` は `c0::rmt/0` となります。ここで、`c0` は、`SCSI` コントローラの論理 `ap_id` であり、`rmt/0` は、`/dev/rmt` 内のテープドライブの論理リンクから生成されます。識別子を `/dev` 内のリンクから生成できない場合は、一意の識別子が割り当てられます。たとえば、テープデバイスが `/dev` 内にリンクを持たない場合、`c0::st3` 形式の `ap_id` が割り当てられます。ここで、`st3` は、内部で生成された一意の識別子です。

コントローラが Solaris I/O マルチパス機能 (旧名 MPxIO または the Sun StorEdge Traffic Manager [STMS]) をサポートしていて、その機能が有効になっている場合は、コントローラによって、scsi_vhci(7D) マルチパスデバイスへのパスが提供されます。このようなコントローラに接続されているデバイスが scsi_vhci(7D) によってサポートされている場合、その `ap_id` は次のようになります。

```
c0::0,0
```

ここで、0,0 は、ターゲットおよび論理ユニットの情報を一意に識別します。このような ap_id のパスの Type フィールドでは、そのパスが scsi_vhci(7D) マルチパスデバイスへのパスを表しているかどうかが、そのパスを通じて接続されているデバイスのタイプとともに示されます。

システム上の接続点の単純な一覧には、SCSI コントローラ上の接続点は含まれますが、SCSI デバイス上の接続点は含まれません。一覧表示オプション (-l) に -a フラグを使用すると、SCSI デバイスも一覧表示されます。例:

```
# cfgadm -l
  Ap_Id              Type         Receptacle     Occupant     Condition
  c0                 scsi-bus     connected      configured   unknown
  sysctrl0:slot0     cpu/mem      connected      configured   ok
  sysctrl0:slot1     sbus-upa     connected      configured   ok
```

SCSI コントローラに加えて SCSI デバイスも一覧表示するには、次のように指定します。

```
# cfgadm -al

Ap_Id                Type         Receptacle   Occupant     Condition
c0                   scsi-bus     connected    configured   unknown
c0::dsk/c0t14d0      disk         connected    configured   unknown
c0::dsk/c0t11d0      disk         connected    configured   unknown
c0::dsk/c0t8d0       disk         connected    configured   unknown
c0::dsk/c0t0d0       disk         connected    configured   unknown
c0::rmt/0            tape         connected    configured   unknown
sysctrl0:slot0       cpu/mem      connected    configured   ok
sysctrl0:slot1       sbus-upa     connected    configured   ok
```

コントローラ c0 で Solaris I/O マルチパス機能が有効になっていて、接続されているディスクデバイスおよびテープデバイスが Solaris I/O マルチパス機能でサポートされている場合、出力は次のようになります。

```
# cfgadm -al

Ap_Id                Type         Receptacle   Occupant     Condition
c0                   scsi-bus     connected    configured   unknown
c0::11,0             disk-path    connected    configured   unknown
c0::14,0             disk-path    connected    configured   unknown
c0::8,0              disk-path    connected    configured   unknown
c0::0,0              disk-path    connected    configured   unknown
c0::a.0              tape-path    connected    configured   unknown
sysctrl0:slot0       cpu/mem      connected    configured   ok
sysctrl0:slot1       sbus-upa     connected    configured   ok
```

接続点の一覧表示については、cfgadm(1M) を参照してください。SCSI コントローラ上の接続ポイントの受容体と占有装置の状態は、次の意味を持ちます。

`empty`
該当しません

`disconnected`
バスは休止しています (バスでの I/O 処理が停止しています)

`connected`
バスはアクティブです

`configured`
バス上に 1 つ以上のデバイスが構成されています

`unconfigured`
バス上にデバイスは構成されていません

個々の SCSI デバイスの対応する状態は次のとおりです。

`empty`
該当しません

`disconnected`
デバイスが接続されているバスは休止しています

`connected`
デバイスが接続されているバスはアクティブです

`configured`
デバイス、またはマルチパス SCSI デバイスへのパスが構成されています

`unconfigured`
デバイス、またはマルチパス SCSI デバイスへのパスは構成されていません

オプション

`cfgadm` では、一覧表示 (`-l`) のほかに数種類の操作が定義されています。これらの操作には、テスト (`-t`)、構成状態の変更の開始 (`-c`)、ハードウェア固有の機能の実行 (`-x`)、および構成管理のヘルプ情報の表示 (`-h`) が含まれます。

`-c` *function*
SCSI ハードウェア固有ライブラリでは、次の汎用コマンドが定義されています。

SCSI コントローラの接続点では、次の構成状態の変更操作がサポートされています。

`connect`
SCSI バスを休止解除します。

`disconnect`
バスを休止します (バスでの I/O 処理を停止します)。

このコマンドの使用法を誤ると、システムがハングアップする場合があります。「注意事項」を参照してください。

`configure`
SCSI バス上に新しいデバイスを構成します。

`unconfigure`
バスに接続されているすべてのデバイスを構成解除します。

SCSI デバイスおよびマルチパス SCSI デバイスへのパスでは、次の汎用コマンドが定義されています。

`configure`
特定のデバイス、またはマルチパス SCSI デバイスへの特定のパスを構成します。

`unconfigure`
特定のデバイス、またはマルチパス SCSI デバイスへの特定のパスを構成解除します。

`-f`
`disconnect` とともに使用すると、ハードウェアでサポートされている場合に、SCSI バスの休止が強制されます。

このコマンドの使用法を誤ると、システムがハングアップする場合があります。「注意事項」を参照してください。

`-h` *ap_id*
ヘルプオプションと任意の SCSI 接続点を組み合わせて使用して、SCSI 固有のヘルプを表示できます。

`-o` *hardware_option*
現在定義されているハードウェア固有のオプションはありません。

`-s` *listing_option*
`select` サブオプションを使用すると、クラス `scsi` の接続点を一覧表示できます。詳細は、`cfgadm(1M)` のマニュアルページを参照してください。

`-t` *ap_id*
現時点では、使用できるテストコマンドはありません。

`-x`*hardware_function*
次のコマンドには、SCSI コントローラでのみ使用できるコマンドと、SCSI デバイスでのみ使用できるコマンドがあります。

次の *controller_ap_id* は、`c0` などの SCSI コントローラの `ap_id` を指します。*device_ap_id* は、 `c0::dsk/c0dt3d0` などの SCSI デバイスの `ap_id` を指します。

次のハードウェア固有の機能が定義されています。

`insert_device` *controller_ap_id*
SCSI コントローラ *controller_ap_id* に新しいデバイスを追加します。

このコマンドは、対話形式でのみ使用するように設計されています。

`remove_device` *device_ap_id*
デバイス *device_ap_id* を削除します。

このコマンドは、対話形式でのみ使用するように設計されています。

`replace_device` *device_ap_id*
デバイス *device_ap_id* を削除し、同じタイプの別のデバイスと交換します。

このコマンドは、対話形式でのみ使用するように設計されています。

`reset_device` *device_ap_id*
device_ap_id をリセットします。

`reset_bus` *controller_ap_id*
バス *controller_ap_id* をリセットします。ただし、そのバスに接続されているデバイスはリセットしません。

`reset_all` *controller_ap_id*
バス *controller_ap_id* と、そのバス上のすべてのデバイスをリセットします。

`locator` [=`on`|`off`] *device_ap_id*
プラットフォームによって提供されている場合に、ハードディスクロケータ LED を設定または取得します。[on|off] サブオプションが設定されていない場合、ハードディスクロケータの状態が出力されます。

`led`[=`LED`,`mode`=`on`|`off`|`blink`] *device_ap_id*
サブ引数が設定されていない場合、この機能によって、現在の LED 設定の一覧が出力されます。サブ引数が設定されている場合、この機能によって、スロットの特定の LED のモードが設定されます。

使用例

例 1 ディスクの構成

次のコマンドは、コントローラ `c0` に接続されているディスクを構成します。

```
# cfgadm -c configure c0::dsk/c0t3d0
```

例 2 ディスクの構成解除

次のコマンドは、コントローラ `c0` に接続されているディスクを構成解除します。

```
# cfgadm -c unconfigure c0::dsk/c0t3d0
```

例 3 新しいデバイスの追加

次のコマンドは、コントローラ `c0` に新しいデバイスを追加します。

```
# cfgadm -x insert_device c0
```

システムは次のように応答します。

例 3 新しいデバイスの追加 *(続き)*

```
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@e,8800000
This operation will suspend activity on SCSI bus c0
Continue (yes/no)?
```

次を入力します。

```
y
```

システムは次のように応答します。

```
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)?
```

次を入力します。

```
y
```

例 4 デバイスの交換

次のコマンドは、コントローラ **c0** に接続されているデバイスを交換します。

```
# cfgadm -x replace_device c0::dsk/c0t3d0
```

システムは次のように応答します。

```
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@e,8800000/sd@3,0
This operation will suspend activity on SCSI bus: c0
Continue (yes/no)?
```

次を入力します。

```
y
```

システムは次のように応答します。

```
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)?
```

次を入力します。

```
y
```

例 5 マウントされたファイルシステムがディスクの構成解除中に検出された場合

次のコマンドは、マウントされたファイルシステムがディスクの構成解除中に検出された場合を示しています。

```
# cfgadm -c unconfigure c1::dsk/c1t0d0
```

システムは次のように応答します。

例 5 マウントされたファイルシステムがディスクの構成解除中に検出された場合 *(続き)*

```
cfgadm: Component system is busy, try again: failed to offline:
/devices/pci@1f,4000/scsi@3,1/sd@1,0
     Resource              Information
------------------  --------------------------
/dev/dsk/c1t0d0s0   mounted filesystem "/mnt"
```

例 6 ディスクのロケータの値の表示

次のコマンドは、ディスクのロケータの値を表示します。この例は、SPARC Enterprise Server ファミリ固有のものです。

```
# cfgadm -x locator c0::dsk/c0t6d0
```

システムは次のように応答します。

```
Disk                    Led
c0t6d0                  locator=on
```

例 7 ディスクのロケータの値の設定

次のコマンドは、ディスクのロケータの値を設定します。この例は、SPARC Enterprise Server ファミリ固有のものです。

```
# cfgadm -x locator=off c0::dsk/c0t6d0
```

システムからは何も出力されません。

例 8 マルチパス SCSI ディスクへのパスの構成

次のコマンドは、コントローラ `c0` を通じて接続されているパスを構成します。

```
# cfgadm -c configure c0::2,0
```

例 9 マルチパス SCSI ディスクへのパスの構成解除

次のコマンドは、コントローラ `c0` を通じて接続されているパスを構成解除します。

```
# cfgadm -c unconfigure c0::2,0
```

ファイル

`/usr/lib/cfgadm/scsi.so.1`
: 汎用的な SCSI ホットプラグ操作のためのハードウェア固有ライブラリ

`/usr/platform/SPARC-Enterprise/lib/cfgadm/scsi.so.1`
: 汎用的な SCSI ホットプラグ操作のためのプラットフォーム固有ライブラリ

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

cfgadm(1M), luxadm(1M), config_admin(3CFGADM), libcfgadm(3LIB), attributes(5), scsi_vhci(7D)

注意事項

disconnect (休止) 操作は、root (/)、/usr、swap、/var などの重要なパーティションが含まれるディスクを制御するコントローラではサポートされていません。このようなコントローラでは、disconnect 操作を実行しないでください。使用法を誤ると、システムがハングアップし、リブートが必要になる場合があります。

コントローラが disconnected (休止) 状態の場合、システムでデッドロックが発生する可能性があります。disconnect 操作を使用するときは、注意するようにしてください。コントローラを disconnected 状態にするのは、DR 操作を実行するために必要な最低限の期間にすることをお勧めします。disconnect コマンドは、システムの実行中に SCSI ケーブルを交換できるようにするためだけに提供されています。このコマンドをほかの目的で使用しないでください。デッドロックが発生した場合の唯一の解決方法は、システムをリブートすることです。

ホットプラグ操作は、すべての SCSI コントローラでサポートされているわけではありません。

警告

一部の SCSI デバイスのコネクタは、SCSI ホットプラグの仕様に準拠していません。このようなデバイスでホットプラグ操作を実行すると、SCSI バス上のハードウェアに障害を与える場合があります。詳細は、ハードウェアのマニュアルを参照してください。

名前

cfgadm_sdcard – cfgadm の SD/MMC ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm  [-f] [-y | -n] [-o hardware_options]
     -c function ap_id[...]

/usr/sbin/cfgadm  [-f] [-y | -n] [-o hardware_options]
     -c hardware_function ap_id[...]

/usr/sbin/cfgadm  [-v] [-a] [-s listing_options]
     [-o hardware_options] [-l [ap_id | ap_type[...]]]

/usr/sbin/cfgadm  [-v] [-o hardware_options] -h [ap_id[...]]
```

機能説明

Secure Digital (SD) および MultiMediaCard (MMC) ハードウェア固有ライブラリ `/usr/lib/cfgadm/sdcard.so.1` は、`cfgadm(1M)` コマンドによる SD/MMC ホットプラグ操作のための機能を提供します。`cfgadm` コマンドは、接続点で実行されます。接続点とは、ハードウェアリソースを動的に再構成できるシステム上の場所です。接続点については、`cfgadm(1M)` を参照してください。

各 SD/MMC スロットは、デバイスツリー上で接続点によって表されます。システム上で接続および構成されているカードデバイスは、接続点の名前の拡張として表されます。このマニュアルページでは、「接続点」および「SD/MMC スロット」という用語は同じ意味で使用されます。

接続点には、ap_id を通じて名前が付けられます。すべての SD/MMC 接続点は、次の形式の文字列で構成されます。

```
sdcardX/[S][::dsk/cXtYd0]
```

各表記の意味は次のとおりです。

X は、SD/MMC コントローラの番号です。

S は、コントローラのスロット番号です (0 - 8)。

`dev/cXtYd0` は、挿入されているメモリーカードを識別します。

Y は、ターゲットの番号です。

一般的に、デバイスの識別子は、`/dev` 内のデバイスの対応する論理リンクから生成されます。SD/MMC デバイスでは 1 つの LUN (LUN 0) しかサポートされないため、デバイス文字列の「d」コンポーネントは常に数値 0 (ゼロ) になります。たとえば、SD/MMC コントローラ 2 のスロット 4 の論理 *ap_id* は、*sdcard2/4* になります。SD/MMC メディアカードがこの接続点に挿入されていて、デバイスが構成されている場合、*ap_id* は *sdcard2/4::dsk/c2t0d0* になります。

デバイスを識別する c*X*t*Y*d*0* 文字列は、デバイスの接続点と 1 対 1 で対応します。

システム上の接続点の単純な一覧には、すべての SD/MMC デバイスのスロットと、接続されているデバイスが含まれます。例:

```
#cfgadm -l
Ap_Id                     Type       Receptacle   Occupant    Condition
 sdcard0/0::dev/c2t0d0    sdcard      connected    configured     ok
 sata0/1::dev/c0t1d0      disk        connected    configured     ok
 sata0/2::dev/c0t2d0      cd-dvd      connected    configured     ok
 sata0/3                  sata-port   empty        unconfigured   ok
 usb0/1                   unknown     empty        unconfigured   ok
 usb0/2                   unknown     empty        unconfigured   ok
```

接続点の一覧表示については、cfgadm(1M) を参照してください。

SD/MMC スロット上の接続点における受容体の状態は、次の意味を持ちます。

empty
: SD/MMC スロットは電源が投入されており、有効になっています。このスロットではデバイスの存在が検出されませんでした。

disconnected
: SD/MMC スロットが有効になっていない、SD/MMC デバイスの存在は検出されたがデバイスとの通信が確立されてない、またはスロットで障害が発生している、のいずれかです。

connected
: スロットで SD/MMC デバイスが検出され、デバイスの通信が確立されています。

占有装置 (SD/MMC スロットに挿入されているデバイス) の状態は、次の意味を持ちます。

configured
: 接続されている SD/MMC デバイスは構成されており、オペレーティングシステムが使用できる状態です。

unconfigured
: デバイスが接続されていないか、SD/MMC スロットに挿入されている SD/MMC デバイスがまだ構成されていません。構成するには、コマンド **cfgadm -c configure ap_id** を実行します。

接続点 (SD/MMC スロット) の条件は、次の意味を持ちます。

ok
: SD/MMC スロットは、電源が投入され、有効になっており、使用できる状態です。

failed
: SD/MMC スロットで障害が発生しています。システムにより無効になっているか電源が切断されている可能性があります。使用不可能であり、その条件は不明です。障害の原因は、スロットに挿入されているデバイスである可能性があります。

unknown
SD/MMC スロットは無効になっており、その条件は不明です。

オプション

cfgadm コマンドでは、一覧表示 (-l) のほかに数種類の操作が定義されています。これらの操作には、構成状態の変更の開始 (-c)、ハードウェア固有の機能の実行 (-x)、および構成管理のヘルプ情報の表示 (-h) が含まれます。

-c: SD/MMC スロットの接続点では、次の構成状態の変更操作がサポートされています。

connect
SD/MMC スロットを有効化 (アクティブ化) して、接続されているデバイスとの通信を確立します。この操作は、必要に応じてスロットに電源を投入することを意味しています。

disconnect
挿入されているデバイスがまだ構成解除されていない場合は構成解除し、SD/MMC スロットを無効化 (非アクティブ化) します。その後に「connect」コマンドを使用した場合、SD/MMC スロット操作は有効になりますが、デバイスは「configured」状態にはなりません。

SD/MMC スロットに挿入された SD/MMC カードでは、次の状態の変更操作がサポートされています。

configure
新しいデバイスがまだ構成されていない場合に、オペレーティングシステムが使用できるように構成します。このコマンドは、必要に応じて接続操作も実行することを意味しています。

unconfigure
SD/MMC スロットに挿入されているデバイスがまだ構成解除されていない場合は、構成解除します。

-f: サポートされていません。

-h *ap_id*: ヘルプオプションと任意の SD/MMC 接続点を組み合わせて使用して、SD/MMC 固有のヘルプを表示できます。

-l [-v]: -l オプションは、cfgadm(1M) で説明されているように機能します。-v オプションと組み合わせて使用した場合、「Information」フィールドには次の SD/MMC 固有の情報が含まれます。

Mod: 製品モデル文字列
Rev: 製品バージョン番号 (メジャー.マイナー)
Date: 製造年月
SN: 製品のシリアル番号 (16 進数)

-o hardware_options - 現在定義されているハードウェア固有のオプションはありません。

-s listing_options: select サブオプションを使用すると、クラス SD/MMC の接続点を一覧表示できます。cfgadm(1M) を参照してください。

-t *ap_id*: セルフテスト機能。SD/MMC スロットではサポートされていません。

-x *hardware_function*: ハードウェア固有の機能を実行します。sdcard_reset_slot *ap_id* は、*ap_id* によって指定された SD/MMC スロットのリセットを示します。スロットに SD/MMC デバイスが挿入されている場合は、そのデバイスもリセットされます。

-v: 詳細モードで実行します。

使用例

例 1 – SD/MMC カードの構成:

次のコマンドは、SD/MMC コントローラ 0、スロット 0 に接続されているカードを構成します。このコマンドは、SD/MMC スロットにデバイスが挿入されている場合にのみ発行してください。

```
# cfgadm -c configure sdcard0/0
```

例 2 – SD/MMC カードの構成解除:

次のコマンドは、SD/MMC コントローラ 0、スロット 3 に挿入されているカードを構成解除します。

```
# cfgadm -c unconfigure sdcard0/3
```

例 3 - マウントされたファイルシステムがディスクの構成解除中に検出された場合:

次のコマンドは、マウントされたファイルシステムがディスクの構成解除中に検出された場合を示しています。

```
# cfgadm -c unconfigure sdcard1/5::dsk/c01t35d0
```

システムは次のように応答します。

```
cfgadm: Component system is busy, try again: failed to offline:
      /devices/pci@0,0/pci8086,244e@1e/pci1095,3124@1/sd@5,0
          Resource              Information
      ------------------  --------------------------
      /dev/dsk/c1t5d0s0   mounted filesystem "/mnt"
```

ファイル

/usr/lib/cfgadm/sdcard.so.1
汎用的な SD/MMC ホットプラグ操作のためのハードウェア固有ライブラリです。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

cfgadm(1M), config_admin(3CFGADM), libcfgadm(3LIB), attributes(5), sda(7D), sdcard(7D), sdhost(7D)

注意事項

通常の操作では、SD/MMC メディアカードは、挿入すると自動的に構成されます。ほとんどの管理者は、通常の使用で、このコマンドを使用する必要はありません。

SD/MMC カードを、最初に構成解除せずに取り外した場合、取り外す際にデバイスに書き込みが行われていたときは、データが失われる可能性があります。読み取り専用でマウントされているデバイスは、いつでも安全に取り外すことができます。

pcfs(7FS) 以外のファイルシステムがマウントされているデバイスは、取り外す前に必ず明示的に構成解除するようにしてください。

名前

cfgadm_shp – cfgadm の PCI Express および標準 PCI ホットプラグ対応ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v]
     [-o hardware_options] -c function ap_id [ap_id]

/usr/sbin/cfgadm [-f] [-y | -n] [-v]
     [-o hardware_options] -x hardware_function ap_id [ap_id]

/usr/sbin/cfgadm [-v] [-s listing_options]
     [-o hardware_options] -x hardware_function ap_id [ap_id]

/usr/sbin/cfgadm [-v] [-o hardware_options] -tap_id [ap_id]

/usr/sbin/cfgadm [-v] [-o hardware_function]-h [ap_id | ap_type]
```

機能説明

PCI Express および標準 PCI ホットプラグ対応ハードウェア固有のライブラリ `/usr/lib/cfgadm/shp.so.1` は、`cfgadm` コマンド (`cfgadm(1M)` を参照) による、ホットプラグ対応システム内の個々のホットプラグ対応スロットに対する、PCI Express および標準 PCI ホットプラグ対応アダプタカードのホットプラグ操作をサポートします。その他の PCI ホットプラグ対応アダプタカード (PCI Express および標準 PCI ホットプラグ対応カード以外) のサポートは、`cfgadm_pci` ライブラリ (`cfgadm_pci(1M)` を参照) によって提供されます。ホットプラグ管理モデルは、このマニュアルページに示されている点を除き、PCI Express ホットプラグおよび標準 PCI ホットプラグで同じです。

PCI ホットプラグの場合、特定の PCI バス上の各ホットプラグスロットは、その PCI バスの接続点によって表されます。

接続点は、受容体と占有装置の 2 つの部分で構成されます。通常、PCI ホットプラグにおける受容体は、物理ホットプラグ対応スロットと見なされ、占有装置は、スロットに接続される PCI アダプタカードと見なされます。

接続点には、`ap_id` を通じて名前が付けられます。`ap_id` には、論理 ap_id と物理 ap_id の 2 種類があります。物理 `ap_id` は、次の例のように物理パス名に基づいています。

```
/devices/pci@7c,0/pci10de,5d@d:pcie2
```

一方、論理 `ap_id` は、`pcie2` などの、より短く、ユーザーにわかりやすい名前です。ホットプラグ PCI の `ap_type` は `pci` です。

`ap_type` は、`Type` フィールドの情報と同じではない点に注意してください。

PCI Express の ap_id の命名

それ自体は PCI Express デバイスではない接続点を含め、PCI Express 階層内に存在する (つまり、親または祖先が PCI Express デバイスである) 接続点では、次に示す名付けスキームが使用されます。

文法:

`APID` : *absolute-slot-path*
　基本項です。

absolute-slot-path : *slot-path*[:*slot-path*[:*slotpath* ...]]
ここで、*fru-id* は、*slot-id* が含まれているシャーシの FRU を示します (ある場合)。

fru-id : *fru-type*[*serialid#*]
ここで、*fru-type* は、PCI Express 拡張シャーシの「iob」であり、その後にそのシリアル番号 *serialid#* が続きます (取得できる場合)。

slot-id : *slot-name* | *device-type physical-slot#* | \
nexus-driver-name nexus-driver-instance.\
device-type pci-device-number
ここで、*slot-name* は、プラットフォームまたはハードウェア自体によって割り当てられる名前です。*device-type* は、PCI Express デバイスを示す `pcie` または PCI デバイスを示す `pci` です。*nexus-driver-name* はデバイスコンポーネントのドライバ名であり、*physical-slot#* はハードウェアスロット番号、*pci-device-number* は標準 PCI 用語における PCI デバイス番号です。

まず、*absolute-slot-path* が構成されます。absolute-slot-path は、接続点の位相上の位置を、ユーザーがより物理的に認識できる用語で表します。この *absolute-slot-path* は、それぞれ : (コロン) で区切られた *slot-path* コンポーネントで構成されます。リーフまたは左端の *slot-path* コンポーネントは、接続点自体のデバイスを表し、右端または最上位の *slot-path* コンポーネントまでのその右隣の *slot-path* コンポーネントは、ルートデバイスまでの親をそれぞれ表します。

各 *slot-path* は、*slot-id* で構成されます。場合によっては、slot-id の前に、*slot-id* によって表されるデバイスが含まれている拡張シャーシを識別する *fru-id* が示されます (後述します)。*fru-id* は、*fru-type* とそれに続くオプションの *serialid#* で構成されます。*fru-type* は、PCI Express 拡張シャーシタイプの「iob」であり、*serialid#* は、拡張シャーシハードウェアから取得される生のシリアル番号を示す 64 ビット 16 進数値か、Sun ブランドの拡張シャーシの場合は、大文字の 4 つの ASCII 文字シーケンスです。

各 slot-id は、次の 3 つのうちのいずれかの形式で構成されます。

slot-id 形式 (1)
slot-names

slot-id 形式 (2)
device-type physical-slot#

slot-id 形式 (3)
nexus-driver-name nexus-driver-instance device-type pci-device-number

選択される形式の優先順位は、形式番号の小さいものから大きなものの順、つまり、上記の上から下の順です。ある形式を正常に構成できない場合は、次に数値の大きい形式が試されます。

slot-id 形式 (1) の *slot-names* は、デバイスツリー上の対応するノードの `slot-names` プロパティーから取得されます。これは、ハードウェアまたはプラットフォームによって割り当てられる名前です。この形式は、事前定義または確立されていません。

slot-id 形式 (2) では、*device-type* は、コンポーネントのスロットのデバイスタイプを示し、PCI Express を示す `pcie` または PCI を示す `pci` です。一方、対応するデバイスノードの `physical-slot#` プロパティーから取得される *physical-slot#* は、コンポーネントのハードウェアスロット番号を示します。

slot-id 形式 (3) が使用されるのは、ほかのすべての形式が正常に構成できないときであり、デフォルトの形式と見なされます。*nexus-driver-name* はコンポーネントのドライバ名、*nexus-driver-instance* はこのドライバのインスタンス、*device-type* は形式 (2) の説明と同じであり、*pci-device-number* は、説明されているとおり PCI デバイス番号であり、標準 PCI 用語におけるデバイス構成サイクルで使用されます。

要約すると、*slot-path* コンポーネントは、その前に示される可能性のあるオプションの FRU コンポーネントを拡張し、次のいずれかの形式で、この順で構成されます。

(1) [iob[*serialid#*].]
slot-names

(2) [iob[*serialid#*].]
device_type physical_slot#

(2) [iob[*serialid#*].]
nexus-driver-name nexus-driver-instance.

device_type pci-device-number

最後に、`cfgadm` で使用される実際の `ap_id` 名の最終的な形式は、次の優先順位で決定されます。

`ap_id` 形式 (1)
absolute-slot-path が `cfgadm` の `ap_id` フィールドの固定長制限内に収まる場合は、*absolute-slot-path* 自体が使用されます。

`ap_id` 形式 (2)
(*absolute-slot-path* が `ap_id` の長さ制限を超える場合) 最後の *slot_path* コンポーネントが拡張シャーシ内に含まれていて、その slot_path に *serialid#* が含まれている場合は、最後の *slot_path* コンポーネントが使用されます。この形式で *serialid#* が必須となるのは、グローバルに一意な `ap_id` にするためです。

`ap_id` 形式 (3)
(*absolute-slot-path* が `ap_id` の長さ制限を超える場合) 最後の *slot_path* コンポーネントのデフォルトの形式、*slot-id* 形式 (3) が使用されます。

どの最終的な ap_id 名が使用されても、*absolute-slot-path* は、-s オプションまたは -v オプションを使用して表示できる情報 (info) フィールドに格納されます。この情報を使用すると、ap_id 形式 (2) または ap_id 形式 (3) を使って名前が付けられた ap_id を物理的に見つけることができます。*absolute-slot-path* は、情報フィールドに格納されるときにわずかに変換されます。具体的には、コロン (:) がスラッシュ (/) に置き換えられ、位相上の文脈がより厳密に示されます。*absolute-slot-path* には、リーフまたは右端の *slot-path* コンポーネントよりも上位でオンボードホストスロットまでの間に、ホットプラグ対応でない *slot-path* コンポーネントを含めることができます。

ホットプラグ対応の例の一覧については、「例」の節を参照してください。

オプション

次のオプションがサポートされています。

-c *function*
: PCI ホットプラグ対応スロットでは、次の機能がサポートされています。

  configure
  : スロット上の PCI デバイスが Solaris で使用されるように構成します。

  connect
  : スロットを PCI バスに接続します。

  disconnect
  : PCI バスからスロットを切断します。

  insert
  : サポートされていません。

  remove
  : サポートされていません。

  unconfigure
  : PCI デバイスのリソースをシステムから論理的に削除します。

-f
: サポートされていません。

-h *ap_id* | *ap_type*
: PCI ホットプラグ固有のヘルプメッセージを表示します。

-l *list*
: PCI ホットプラグスロットの値を一覧表示します。

-o*hardware_options*
: 現在定義されているハードウェア固有のオプションはありません。

-s *listing_options*
: 汎用の cfgadm(1M) と同じです。

-t *ap_id*
このコマンドは、スロットでテスト機能をサポートするプラットフォームでのみサポートされます。

-v
詳細モードで実行します。

-v オプションを -l オプションとともに使用すると、cfgadm コマンドによって、接続点に関する情報が出力されます。PCI Express 階層内に存在する接続点の場合、情報フィールドに、接続点の絶対スロットパスの位置が示されます。これには、スロットパスに含まれる各コンポーネントのハードウェアまたはプラットフォーム固有のラベル情報が含まれます。スロットパスに含まれる各コンポーネントは、/ (スラッシュ) で区切られます。上記の「PCI Express の ap_id の命名」を参照してください。PCI Express 階層内に存在しない PCI ホットプラグの接続点については、cfgadm_pci(1M) を参照してください。Type フィールドの情報は、-v オプションを指定したかどうかにかかわらず出力されます。占有装置の Type フィールドは、スロットの内容を示します。次の 2 種類の値があります。

unknown
スロットは空です。スロットにカードが装着されている場合、カードが構成されていないか、カード上のデバイスのドライバがありません。

subclass/*board*
スロットに装着されているカードは、シングルファンクションデバイスまたはマルチファンクションデバイスです。

subclass は、デバイスのサブクラスコードを表す文字列であり、SCSI、ethernet、pci-isa などが示されます。カードがマルチファンクションデバイスである場合、代わりに MULT と表示されます。

board は、デバイスのボードタイプを表す文字列です。たとえば、hp は、PCI ホットプラグアダプタの場合に使用される文字列です。

-x*hardware_function*
ハードウェア固有の機能を実行します。通常は、これらのハードウェア固有の機能を使用して、受容体または占有装置の状態を変更しないでください。

次の *hardware_function* がサポートされています。

led=[*led_sub_arg*],mode=[*mode_sub_arg*]
サブ引数を指定しない場合、現在の LED 設定の一覧が表示されます。サブ引数を指定した場合、スロットの特定の LED のモードが設定されます。

led_sub_arg は、fault、power、attn、または active に指定します。

mode_sub_arg は、on、off、または blink に指定します。

PCI Express の場合は、power および attn LED のみが有効であり、attn LED の状態のみを変更できます。

LED の状態を変更しても、受容体または占有装置の状態は変更されません。通常は、LED はホットプラグコントローラによって制御され、ユーザーの操作は不要です。このコマンドは、テストのために使用します。

注意 – LED の状態を変更すると、占有装置または受容体の状態が誤って示される場合があります。

次のコマンドは、LED の値を表示します。

```
example# cfgadm -x led pcie2
Ap_Id             Led
pcie2     power=on,fault=off,active=off,attn=off
```

次のコマンドは、attn LED が点滅してスロットの位置を示すように設定します。

```
example# cfgadm -x led=attn,mode=blink pcie2
```

使用例

例 1 各スロットの値の表示

次のコマンドは、各スロットの値を表示します。

```
example# cfgadm -l
Ap_Id            Type         Receptacle   Occupant       Condition
c0               scsi-bus     connected    configured     unknown
c1               scsi-bus     connected    unconfigured   unknown
c2               scsi-bus     connected    unconfigured   unknown
pcie7            etherne/hp   connected    configured     ok
pcie8            unknown      empty        unconfigured   unknown
pcie9            fibre/hp     connected    configured     ok
```

例 2 カードの交換

次のコマンドは、すべての DR 対応の接続点を一覧表示します。

```
example# cfgadm
Type             Receptacle   Occupant     Condition
c0               scsi-bus     connected    configured     unknown
c1               scsi-bus     connected    unconfigured   unknown
c2               scsi-bus     connected    unconfigured   unknown
pcie7            etherne/hp   connected    configured     ok
pcie8            unknown      empty        unconfigured   unknown
pcie9            fibre/hp     connected    configured     ok
```

次のコマンドは、pcie7 で識別されるカードを構成解除し、電気的に切断します。

```
example# cfgadm -c disconnect pcie7
```

次のコマンドを入力して変更を確認できます。

例 2　カードの交換　　(続き)

```
example# cfgadm pcie7
Ap_Id     Type         Receptacle   Occupant     Condition
pcie7     unknown      disconnected unconfigured unknown
```

この時点で、カードを交換できます。次のコマンドは、交換用カードを電気的に接続し、構成します。

```
example# cfgadm -c configure pcie7
```

次のコマンドを入力して変更を確認できます。

```
example# cfgadm pcie7
Ap_Id     Type         Receptacle   Occupant     Condition
pcie7     etherne/hp   connected    configured   ok
```

例 3　PCI Express トポロジにおける ApId の解釈

次のコマンドは、ホストレベルでホットプラグ対応スロットに接続されている I/O 拡張シャーシ内に PCI Express および PCI 接続点の両方があるトポロジの一覧を示します。

```
example# cfgadm -s cols=ap_id:info
Ap_Id                          Information
iou#0-pci#0                    Location: iou#0-pci#0
iou#0-pci#1                    Location: iou#0-pci#1
iou#0-pci#1:iob.pci3           Location: iou#0-pci#1/iob.pci3
iou#0-pci#1:iob.pci4           Location: iou#0-pci#1/iob.pci4
iou#0-pci#2                    Location: iou#0-pci#2
iou#0-pci#2:iob58071.pcie1     Location: iou#0-pci#2/iob58071.pcie1
iou#0-pci#2:iob58071.special   Location: iou#0-pci#2/iob58071.special
iou#0-pci#3                    Location: iou#0-pci#3
iou#0-pci#3:iobBADF.pcie1      Location: iou#0-pci#3/iobBADF.pcie1
iou#0-pci#3:iobBADF.pcie2      Location: iou#0-pci#3/iobBADF.pcie2
iou#0-pci#3:iobBADF.pcie3      Location: iou#0-pci#3/iobBADF.pcie3
iou#0-pci#3:iobBADF.pci1       Location: iou#0-pci#3/iobBADF.pci1
iou#0-pci#3:iobBADF.pci2       Location: iou#0-pci#3/iobBADF.pci2
```

この例では、`iou#0-pci#[0-3]` エントリは、システム上の最上位のホットプラグ対応スロットを表しています。`iou#`*n*`-pci#`*n* の形式は、上記の文法の仕様の節で示されているいずれの形式とも一致しないため、このホットプラグトポロジの基本コンポーネントのこのような名前は、`slot-names` プロパティーを通じてプラットフォームから取得されたものであると推測できます。

前の出力に示されているスロットの説明を次に示します。

スロット `iou#0-pci#0`
　このスロットは空であるか、その占有装置が構成されていません。

例 3　PCI Express トポロジにおける ApId の解釈　　(続き)

スロット iou#0-pci#1
　このスロットには、2 つのホットプラグ対応スロット pci3 と pci4 がある拡張シャーシが含まれています。pci3 と pci4 は、その拡張シャーシ内に含まれている、それぞれ物理スロット番号 3 と 4 を持つ 2 つの PCI スロットを表しています。この例では、拡張シャーシにシリアル ID がないか、エクスポートされていません。

スロット iou#0-pci#2
　このスロットには、16 進数のシリアル ID 58071 を持つサードパーティーの拡張シャーシが含まれています。その拡張シャーシ内には、2 つのホットプラグ対応スロット pcie1 と special があります。pcie1 は、物理スロット番号 1 を持つ PCI Express スロットを表しています。スロット special には、プラットフォーム、ハードウェア、またはファームウェアから取得されたラベルがあります。

スロット iou#0-pci#3
　このスロットには、FRU 識別子 BADF を持つ Sun 拡張シャーシが含まれています。この拡張シャーシには、それぞれ物理スロット番号 1、2、および 3 を持つ 3 つの PCI Express スロット pcie1、pcie2、および pcie3 と、それぞれ物理スロット番号 1 および 2 を持つ 2 つの PCI スロット pci1 と pci2 が含まれています。

次のコマンドは、接続されたホットプラグ対応およびホットプラグ非対応のホストスロットがある I/O 拡張シャーシ内に PCI Express および PCI 接続点の両方があるトポロジの一覧を示します。

```
example# cfgadm -s cols=ap_id:info
Ap_Id                          Information
Slot1                          Location: Slot1
Slot2:iob4ffa56.pcie1          Location: Slot2/iob4ffa56.pcie1
Slot2:iob4ffa56.pcie2          Location: Slot2/iob4ffa56.pcie2
Slot5:iob3901.pci1             Location: Slot2/iob3901.pci1
Slot5:iob3901.pci2             Location: Slot2/iob3901.pci2
```

この例では、ホストシステムには、ホットプラグ対応スロット Slot1 のみがあります。Slot2 および Slot5 は、cfgadm でそれ自体が接続点として表示されていないため、これらはホットプラグ対応でないと推測できます。ただし、Slot2 と Slot5 には、それぞれ、ホットプラグ対応スロットがあるサードパーティーの拡張シャーシが含まれています。

次のコマンドは、特定のデバイスプロパティーがない接続点があるトポロジの一覧を示します。

```
example# cfgadm -s cols=ap_id:info
Ap_Id                          Information
px_pci7.pcie0                  Location: px_pci7.pcie0
px_pci11.pcie0                 Location: px_pci11.pcie0
```

例 3　PCI Express トポロジにおける ApId の解釈　　*(続き)*

```
px_pci11.pcie0:iob.pcie1       Location: px_pci11.pcie0/iob.pcie1
px_pci11.pcie0:iob.pcie2       Location: px_pci11.pcie0/iob.pcie2
px_pci11.pcie0:iob.pcie3       Location: px_pci11.pcie0/iob.pcie3
```

この例では、ホストシステムに 2 つのホットプラグ対応スロット px_pci7.pcie0 と px_pci11.pcie0 が含まれています。ここでは、より優先度が高くわかりやすいほかの形式を構成するための十分な情報をフレームワークが取得できなかったため、*absolute-slot-path* 内の基本 *slot-path* コンポーネントの slot-id 形式 (3) (デフォルトの形式) が使用されています。

右から左に解釈すると、接続点 px_pci7.pcie0 は、nexus ドライバ px_pci、インスタンス 7 にバインドされた、PCI デバイス番号が 0 (同じ番号の物理スロット番号という意味はありません) の PCI Express スロットを表しています。同様に、接続点 px_pci11.pcie0 は、px_pci のドライバインスタンス 11 にバインドされた、PCI デバイス番号が 0 の PCI Express スロットを表しています。

px_pci11.pcie0 の下位には、シリアル ID のない、3 つのホットプラグ対応 PCI Express スロットを持つサードパーティーの拡張シャーシがあります。

次のコマンドは、ApId フィールドの長さ制限を超えている接続点のパスがあるトポロジの一覧を示します。

```
example# cfgadm -s cols=ap_id:info
Ap_Id                          Information
pcie4                          Location: pcie4
pcie4:iobSUNW.pcie1            Location: pcie4/iobSUNW.pcie1
pcie4:iobSUNW.pcie2            Location: pcie4/iobSUNW.pcie2
iob8879c3f3.pci1
                 Location: pcie4/iobSUNW.pcie2/iob8879c3f3.pci1
iob8879c3f3.pci2
                 Location: pcie4/iobSUNW.pcie2/iob8879c3f3.pci2
iob8879c3f3.pci3
                 Location: pcie4/iobSUNW.pcie2/iob8879c3f3.pci3
```

この例では、ホストにホットプラグ対応スロット pcie4 のみが存在しています。pcie4 の下位には、FRU 識別子が SUNW の Sun 拡張シャーシが接続されています。その拡張シャーシの PCI Express スロット pcie2 (ApId pcie4:iobSUNW.pcie2) の下位には、3 つのホットプラグ対応 PCI スロットがある別の拡張シャーシが入れ子になっています。

次の absolute-slot-path 形式の長さが、

```
pcie4/iobSUNW.pcie2/iob8879c3f3.pci1...3
```

...ApId フィールドの長さ制限を超えており、リーフ slot-path コンポーネントがグローバルに一意であるため、*absolute-slot-path* 内のリーフ *slot-path* コンポーネントが最終的な ApId として使用される ap_id 形式 (2) が使用されています。

例 3　PCI Express トポロジにおける ApId の解釈　　(続き)

次のコマンドは、ApId フィールドの長さ制限を超えていて、(シリアル ID が見つからないなど) リーフ *slot-id* だけで一意に識別できるだけの十分な情報がない接続点のパスがあるトポロジの一覧を示します。

```
example# cfgadm -s cols=ap_id:info
Ap_Id                          Information
pcie4                          Location: pcie4
pcie4:iob4567812345678.pcie3   Location: pcie4/iob4567812345678.pcie3
px_pci20.pcie0
                Location: pcie4/iob4567812345678.pcie3/iob.pcie1
px_pci21.pcie0
                Location: pcie4/iob4567812345678.pcie3/iob.pcie2
```

この例では、ホストにホットプラグ対応スロット pcie4 のみが存在しています。pcie4 の下位には、16 進数のシリアル ID が 4567812345678 のサードパーティーの拡張シャーシが接続されています。その拡張シャーシの PCI Express スロット pcie3 (ApId pcie4:iob4567812345678.pcie3) の下位には、シリアル ID がなく、2 つのホットプラグ対応 PCI Express スロットがある別のサードパーティーの拡張シャーシが入れ子になっています。

次の absolute-slot-path 形式の長さが、

```
pcie4/iob4567812345678.pcie3/iob.pcie1...2
```

ApId フィールドの長さ制限を超えており、リーフ *slot-path* コンポーネントがグローバルに一意ではないため、ap_id 形式 (3) が使用されています。ap_id 形式 (2) では、*absolute-slot-path* 内のリーフ *slot-path* コンポーネントの *slot-id* 形式 (3) (デフォルトの形式) が最終的な ApId として使用されます。

リーフコンポーネント .../iob.pcie1 のデフォルトの形式、つまり slot-id 形式 (3) は、px_pci のドライバインスタンス 20 にバインドされた、デバイス番号 0 の PCI Express スロットを表しています。同様に、リーフコンポーネント .../iob.pcie2 のデフォルトの形式は、px_pci のドライバインスタンス 21 にバインドされた、デバイス番号 0 の PCI Express スロットを表しています。

ファイル

/usr/lib/cfgadm/shp.so.1
: PCI Express および標準 PCI ホットプラグ操作のためのハードウェア固有のライブラリです。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library
インタフェースの安定性	不確実

関連項目

cfgadm(1M), cfgadm_pci(1M), hotplugd(1M), config_admin(3CFGADM), libcfgadm(3LIB), attributes(5), smf(5)

注意事項

cfgadm_shp ライブラリは、smf(5) によって、FMRI の下で管理される hotplug サービスに依存しています。

```
svc:/system/hotplug:default
```

cfgadm_shp ライブラリを適切に機能させるには、サービスを有効にする必要があります。詳細は、hotplugd(1M) を参照してください。

名前

cfgadm_sysctrl – EXX00 システムボードの管理

形式

```
/usr/sbin/cfgadm -c function [-f]
     [-o disable-at-boot | enable-at-boot] [-n | -y] sysctrl0:slot# ...

/usr/sbin/cfgadm -x quiesce-test sysctrl0:slot#

/usr/sbin/cfgadm -x insert-test | remove-test sysctrl0:slot# ...

/usr/sbin/cfgadm -x set-condition-test=# sysctrl0:slot# ...

/usr/sbin/cfgadm [-l]
     -o disable-at-boot | enable-at-boot sysctrl0:slot# ...
```

機能説明

sysctrl ハードウェア固有ライブラリ、/usr/platform/sun4u/lib/cfgadm/sysctrl.so.1 により、E6X00、E5X00、E4X00、E3X00 システム上で、動的再構成を用いてシステムボードの構成や構成解除ができるようになります。これによりシステムが稼働中であってもリブートせずに、I/O ボード、および CPU ボードを、Solaris 用に設定されたシステムのスロットに挿入できるようになります。また同様に、いずれのタイプのボードも稼働中のシステムから切断および取り外しが可能になり、リブートの必要はなくなります。

システムスロットはデバイスツリー上で「接続点 (attachment point)」として表現され、システムシャーシ内の各スロットには、この接続点が 1 つづつ設定されます。スロットにボードが設置されていない場合は、受容体の状態は empty と認識されます。ボードの電源が切断されシステムから取り外せる状態の場合は、受容体の状態は disconnected (接続解除) と認識されます。ボードに電源が投入されシステムバスに接続されている状態であれば、受容体の状態は connected (接続) と認識されます。

受容体の状態が empty または disconnected の場合、その受容体の占有装置は unconfigured の状態にあると認識されます。受容体の状態が connected の場合は、その受容体の占有装置は configured またはunconfigured のいずれかの状態を取ります。

Solaris でボード上のデバイスを利用できるのは、この状態が configured になっている場合です。unconfigured になっている場合、ボード上のデバイスは利用できません。

ボードを挿入すると、受容体の状態は empty から disconnected に変わります。ボードを取り外すと、受容体の状態は disconnected から empty に変わります。connected になっているボードを取り外してしまうと、オペレーティングシステムのクラッシュを起こし、システムに修復不可能な障害を与える場合があります。

オプション

オプションについての詳細は、cfgadm(1M) の項目を参照してください。

次のオプションがサポートされています。

-c *function*

状態の変更に使用します。*function* の部分に connect 、disconnect 、configure 、unconfigure のいずれかを指定します。

configure

占有装置の状態を configure に変更します。

受容体の状態が disconnected になっている場合は、configure 機能はまず受容体への接続を試みます。connect 機能の一部として作成される OBP デバイスツリーを調べて Solaris デバイスノードを作成し、要求のあったデバイスを接続します。CPU/メモリーボードの場合は、電源が切断状態にある CPUのリストに、この CPU を加えます。この情報は psrinfo(1M) または psradm(1M) コマンドを用いて見ることができます。CPU/メモリーボードにはメモリー接続点が 2 つ作成されます。新規に設置したボード上の I/Oデバイスを使えるように設定するには、mount(1M) および ifconfig(1M) コマンドを利用します。CPU を使用するには、psradm -n を使用して新たなプロセッサをオンラインに設定します。メモリーバンクのテストおよび設定には cfgadm_ac(1M) を利用します。

connect

受容体の状態を connected に変更します。

受容体の状態の変更を要求すると、バス信号の接続中はシステムバスを凍結し、その間にボードをテストします。休止処理によってバスは凍結しま

す。この休止処理の間は、すべてのプロセス活動は停止し、すべてのドライバの動作が中断します。この休止処理およびそれに後続する復元処理にはかなりの時間がかかります。またすべてのドライバがこの機能をサポートしているわけではありません。このため、`-x quiesce-test` というオプションが用意されています。これを使用してシステムバスを凍結させている間に、接続されているボードをファームウェアからテストできます。I/O ボードの場合はこの処理には短時間しか費やしませんが、CPU/メモリーボードの場合は CPU 外部キャッシュのテストが原因で非常に長時間かかります。ここではメモリーのテストはしません。休止処理を開始する前には、実行の確認を促してきます。オプションに `-y` または `-n` を使用すると、この確認作業を省略できます。`disabled-at-boot`に指定されているボードは、接続処理を拒否します。ただし、指定された処理を強制実行する `-f` フラグを利用、または `-o enable-at-boot` を使って指定を無効化した場合は、接続処理が実行されます。`-l` の解説を参照してください。

disconnect　受容体の状態を `disconnected` に変更します。

占有装置の状態が `configure` になっている場合は、まず占有装置を構成解除します。この `disconnect` では休止処理を実行しないので、短時間で終了しま

す。ボードの電源は切断され、取り外しが可能な状態になります。

unconfigure　占有装置の状態を unconfigureed に変更します。

この処理の間、ボード上のデバイスはSolaris からは見えない状態になります。I/O ボード上の I/O デバイスは、Solaris のデバイスツリーから削除されます。デバイスが使用中の場合、この unconfigure の処理は停止し、使用中であることを報告してきます。この場合はデバイスの使用を停止して、その後で unconfigure 処理を再試行する必要があります。CPU/メモリーボードの場合は、ボードの unconfigure 処理を実行する前に、メモリーの状態を unconfigure に設定しておかなければいけません。ボード上の CPU をオフラインにしてから、電源を切断して Solaris CPU リストから削除します。接続中のプロセスが存在する CPU をオフラインにすることはできません。CPU のオフライン化についての詳細は、psradm(1M)、psrinfo(1M)、pbind(1M の各項目を参照してください。

-f　強制的に、ボードに接続されているブロックの非揮発性変数 disabled-board-list を、disabled-at-boot に指定します。『特記事項: Sun Enterprise 6x00、5x00、4x00、3x00 システム』のマニュアルも参照してください。

-l　リストを表示させるオプションです。これはcfgadm(1M) で解説しているものと同様の機能です。

type フィールドには cpu/mem、mem、dual-sbus、sbus-upa、dual-pci、soc+sbus、s のいずれかを指定できます。

ハードウェア固有情報のフィールドは、次のように設定されています。 [disabled at boot] [non-detachable] [100 MHz capable]

ボードのタイプが sbus-upa または soc+upa の場合は、まず最初に次のような追加情報が表示されます。 [single buffered ffb|double buffered ffb|no ffb installed] ボードのタイプが disk の場合は、次のような追加情報が最初に表示されます。 {target: # | no disk} {target: # | no disk}

-o disable-at-boot | enable-at-boot

非揮発性変数 disabled-board-list の変更に使用します。この -o オプションを使う時は、-c *function* または -l オプションと併用します。

disabled-at-boot に設定されているボードに関連するブロックをブート時に有効となるようにするには、 -o enable-at-boot と -c connect を併用します。

-x insert-test | remove-test

テストを実行します。

ここで remove-test を使うと、ボードを物理的に取り外さなくても、テストシーケンスを自動処理している間、指定したスロットのドライバ状態は disconnected から empty に変更されます。

また、insert-test の方を使った場合は、remove-testコマンドで empty に指定したスロットのドライバ状態が isconnected に変更され、ボードはスロットに挿入されているものとして認識されます。

-x quiesce-test sysctrl0:slot1

テストを実行します。

このテストの実行により、ボードの接続処理に必要となる quiesce 処理が実行できる状態になります。現在のソフトウェアおよびハードウェアの構成を保ったままで、システムを確実に休止できるようになります。休止処理できないデバイスやプロセスが存在した場合は、エ

ラーメッセージでその名前を表示します。このコマンドは有効なボード接続点に対してであればどれにでも使用可能ですが、いかなるシステムも必ず slot1 を 1 つ所有しているので、上に示した形式での使用を推奨しておきます。

-x set-condition-test=#

テストを実行します。

状態変更コマンドのポリシーロジックをテストできるように、システムボードの接続点の条件を設定します。新しい設定の内容は、1 から 4 の数字で表現されます。各数字の意味は、以下のようになります。

```
0       unknown
1       ok
2       failing
3       failed
4       unusable
```

オペランド

次のオペランドがサポートされています。

sysctrl0:slot#

`sysctrl` ドライバのインスタンス `0` (`sysctrl0`) が、`EXX00` システム上のボードに接続点を発行します。接続点の名前には、`slot0` から `slot15` までの番号を割り当てます。`#` の部分には `0` から `15` までの数字を入れます。この値がスロット番号を示すことになります。この形式は、`cfgadm(1M)` を用いた論理的な `ap_id` の指定と一致します。これに対応する物理的な `ap_id` は、次項の【ファイル】に記載してあります。

ファイル

/usr/platform/sun4u/lib/cfgadm/sysctrl.so.1
ハードウェア固有ライブラリ

/devices/central@1f,0/fhc@0,f8800000/clock-board@0,900000:slot*
接続点

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library/platform

関連項目

`cfgadm(1M)`, `cfgadm_ac(1M)`, `ifconfig(1M)`, `mount(1M)`, `pbind(1M)`, `psradm(1M)`, `psrinfo(1M)`, `config_admin(3CFGADM)`, `attributes(5)`

『Sun Enterprise 6x00、5x00、4x00、3x00 システム Dynamic Reconfiguration ユーザーマニュアル』

『特記事項: Sun Enterprise 6x00、5x00、4x00、3x00 システム』

注意事項

EXX00 システムの CPU/メモリーボードの動的再構成について、更に詳しい情報が必要な場合は、『Sun Enterprise 6x00、5x00、4x00、3x00 システム Dynamic Reconfiguration ユーザーマニュアル』を参照してください。

名前

cfgadm_usb – cfgadm の USB ハードウェア固有のコマンド

形式

```
/usr/sbin/cfgadm [-f] [-y | -n] [-v] -c function ap_id...

/usr/sbin/cfgadm -f [-y | -n] [-v] [-o hardware_options]
     -x hardware_function ap_id...

/usr/sbin/cfgadm -v [-a] [-s listing_option]
     [-l [ap_id | ap_type...]]

/usr/sbin/cfgadm -v -h [ap_id]...
```

機能説明

ユニバーサルシリアルバス (USB) ハードウェア固有ライブラリ `/usr/lib/cfgadm/usb.so.1` は、`cfgadm`(1M) コマンドを介して USB デバイスを管理するための機能を提供します。`cfgadm` は、接続点で実行されます。接続点の詳細は、`cfgadm(1M)` を参照してください。

USB の管理では、サポートされる接続点は、USB バスに接続されたハブのポートのみです。

接続点には、接続点 ID (*ap_id*) を通じて名前が付けられます。USB バスは階層構造であるため、*ap_id* も同様です。USB ハブには、1 から *n* までの番号が付けられたポートがあります。すべての USB の *ap_id* は、次の形式の文字列で構成されます。

usb*N*/*A*[.*B*[.*C*[...]]]

各情報の意味は次のとおりです。

N は、システム上の *N* 番目の USB ホストコントローラを示し、
A は、ルート (最上位) ハブのポート #*A* を示します。
B は、上位のハブのポート #*A* に接続されたハブのポート #*B* を示します。
C は、上位のハブのポート #*B* に接続されたハブのポート #*C* を示します。以降も同様に続きます。

たとえば、USB コントローラ 0 (唯一のコントローラ) のルートハブの 1 番目のポートは、次のような論理 *ap_id* を持ちます。

```
usb0/1
```

同様に、1 番目の USB コントローラのルートハブの 1 番目のポートに接続された 1 番目の外部ハブの 2 番目のポートは、次のような論理 *ap_id* を持ちます。

```
usb0/1.2
```

たとえば、*ap_id* が usb0/1.4.3.4 である場合は、システム上の 1 番目の USB ホストコントローラのルートハブのポート 1 に接続されたハブのポート 4 に接続されたハブのポート 3 に接続されたハブのポート 4 を表しています。

```
example# cfgadm -l
Ap_Id                Type         Receptacle   Occupant     Condition
usb0/1               USB-hub      connected    configured   ok
```

```
usb0/2               unknown      empty        unconfigured ok
usb0/1.1             USB-storage  connected    configured   ok
usb0/1.2             unknown      empty        unconfigured ok
usb0/1.3             unknown      empty        unconfigured ok
usb0/1.4             USB-device   connected    configured   ok
```

USB2.0 チップには、EHCI ホスト USB2.0 ホストコントローラが 1 つと、いくつかのコンパニオン USB 1.*x* ホストコントローラ (OHCI または UHCI ホストコントローラ) が搭載されています。

USB2.0 デバイスは、接続されると、EHCI 論理ポートに接続されます。これらの論理ポートは、システム上の外部の物理ポート番号に対して 1 対 1 のマッピングになっていない場合があります。USB1.*x* デバイスを接続すると、EHCI ホストコントローラは、デバイスにコンパニオンホストコントローラの別の経路を指定し、デバイスはコンパニオンの論理ポート番号に接続されます。

論理ポート番号と物理ポート番号のマッピングは非常に複雑になる場合があります。例:

```
% cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
c0                   scsi-bus     connected    configured   unknown
usb0/1               usb-mouse    connected    configured   ok
usb0/2               usb-kbd      connected    configured   ok
usb0/3               unknown      empty        unconfigured ok
usb0/4               usb-hub      connected    configured   ok
usb0/4.1             unknown      empty        unconfigured ok
usb0/4.2             unknown      empty        unconfigured ok
usb0/4.3             unknown      empty        unconfigured ok
usb0/4.4             usb-storage  connected    configured   ok
usb1/1               unknown      empty        unconfigured ok
usb1/2               unknown      empty        unconfigured ok
usb1/3               unknown      empty        unconfigured ok
usb2/1               unknown      empty        unconfigured ok
usb2/2               usb-device   connected    configured   ok
usb3/1               unknown      empty        unconfigured ok
usb3/2               unknown      empty        unconfigured ok
usb3/3               unknown      empty        unconfigured ok
usb3/4               unknown      empty        unconfigured ok
usb3/5               unknown      empty        unconfigured ok
```

この例では、usb0 はオンボード USB 1.*x* ホストコントローラです。usb1 と usb2 は、コンパニオン OHCI USB1.*x* ホストコントローラであり、usb3 は EHCI USB2.0 ホストコントローラです。

次の表は、この USB2.0 チップのややわかりにくい経路指定を示しています。

```
logical port number        physical port number
-------------------        --------------------
    usb1/1            internal port 1
```

```
usb1/2            external port 1
usb1/3            external port 3

usb2/1            internal port 2
usb2/2            external port 2

usb3/1            internal port 1
usb3/2            internal port 2
usb3/3            external port 1
usb3/4            external port 2
usb3/5            external port 3
```

残念ながら、多くの場合、正確な経路指定は実験によってしかわかりません。

USB ポートにおける接続点の受容体の状態は、次の意味を持ちます。

`connected`
: USB は、電源が投入され、有効になっています。USB デバイスがポートに接続されています。デバイスが USB バスに論理的に接続されています。

`disconnected`
: USB は、電源が投入され、有効になっています。USB デバイスがポートに接続されています。(`cfgadm -c disconnect` コマンドを使用して) デバイスが USB バスから論理的に切断されています。

`empty`
: USB ポートは電源が投入されていますが、デバイスが接続されていません。

USB ポートにおける USB ポートの接続点でのデバイスの占有装置の状態は、次の意味を持ちます。

`configured`
: USB ポートの USB デバイスは構成されており、Solaris で使用できます。

`unconfigured`
: USB ポートの USB デバイスは、`cfgadm -c unconfigure` を使用して明示的にオフラインにされているか、ドライバがない、またはデバイスに問題があるなど、Solaris で使用できるように正常に構成されていません。

接続点の条件は次のとおりです。

`ok`
: 通常の状態 - 使用できます。

`failing`
: 使用されていません。

`failed`
: 使用されていません。

`unusable`
: アプリケーションによってデバイスが開かれているときに、ユーザーによりデバイスが物理的に取り外されました (未処理の `I/O` がある場合があります)。ユーザーは、同じ物理デバイスを再度挿入し、アプリケーションを適切に終了してから、デバイスをあらためて取り外す必要があります。この処理が完了するまで、ほかの挿入されたデバイスをポートで構成することはできません。

  元のデバイスをポートに再度挿入できない場合は、『Oracle Solaris の管理: 一般的なタスク』に記載されている、この接続点の条件をクリアする手順を参照してください。

`unknown`
: 使用されていません。

USB デバイスは、いつでもホットプラグで接続または取り外しでき、システムによってイベントが検出され、適切に処理されます。

デバイスを USB から取り外す前に、受容体を `disconnected` 状態に移行する必要はありません。ただし、使用中のデバイス (ボリュームマネージャーまたはその他のアプリケーションによって現在開かれているリムーバブルディスクなど) をホットリムーブすることはお勧めしません。

オプション

`cfgadm` では、数種類の操作が定義されています。これらの操作には、構成状態の変更の開始 (`-c`)、ハードウェア固有の機能の実行 (`-x`)、および構成管理のヘルプ情報の表示 (`-h`) が含まれます。

いずれかの操作が失敗した場合、デバイスおよび接続点が想定した状態にならない場合があります。デバイスの現在のステータスを表示するには、`cfgadm -l` コマンドを使用します。

その他のすべてのオプションは、`cfgadm(1M)` での定義と同じ意味を持ちます。

次のオプションがサポートされています。

`-c` *function*
: USB ハードウェア固有ライブラリでは、次の汎用コマンドが定義されています。次の構成状態の変更操作がサポートされています。

  `configure`
  : ポートに接続されている USB デバイスがある場合、このコマンドにより、Solaris で使用できるようにそのデバイスの構成とすべての設定が試みられます。このコマンドでは、必要に応じて暗黙的に、`connect` (`disconnect` の逆) が実行されます。このポートのデバイスがすでに構成されている場合、このコマンドは何も実行せず、エラーメッセージを返します。このコマンドが正常に実行されると、デバイスを Solaris で使用できるようになります。

disconnect
: *ap_id* で unconfigure を実行し (まだ unconfigured になっていない場合)、受容体を disconnected 状態に移行します。これは、デバイスがまだポートに接続されていても実行されます。ドライバを見つけることができ、デバイスの列挙と構成に問題がないと仮定して、cfgadm -c configure を発行するか、デバイスを物理的にホットプラグ接続すると、デバイスでは、受容体の状態が connected に、占有装置の状態が configured に戻ります。

unconfigure
: ポートに接続されているデバイスを Solaris で使用できないようにします (オフラインにします)。成功した場合、cfgadm により、この *ap_id* の占有装置の状態は *unconfigured* と報告されます。configure を *ap_id* に発行すると (成功した場合)、デバイスをポートに物理的にホットプラグ接続した場合と同様に、その占有装置の条件は configured (オンライン) に戻ります。

-f
: サポートされていません。

-h *ap_id*
: ヘルプオプションと任意の USB 接続点を組み合わせて使用して、USB 固有のヘルプを表示できます。

-l[v]
: -l オプションは、cfgadm(1M) で説明されているように機能します。-v オプションと組み合わせて使用すると、Information フィールドに次の USB 固有の情報が表示されます。

- Mfg: 製造元文字列 (iManufacturer)
- Product: 製品文字列 (iProduct)
- NConfigs: デバイスでサポートされる構成の合計数 (bNumConfigurations)
- Config: 10 進数の現在の構成設定 (構成値ではなく、構成インデックス)
- 現在の構成の構成文字列記述子 (iConfiguration)

これらのフィールドの説明については、USB の仕様を参照してください。

-o*hardware_options*
: ハードウェアオプションは、ハードウェア固有のコマンド -x usb_config でのみサポートされます。使用できるオプションの説明については、下記のそのコマンドの説明を参照してください。

-s*listing_options*
: select サブオプションを使用すると、クラス USB の接続点を一覧表示できます。cfgadm(1M) を参照してください。

-x*hardware_function*
: 次のハードウェア固有の機能が定義されています。

usb_config -o config=*n*
: このコマンドでは、-o オプションを使用して必須の config 値を指定する必要があります。

ap_id の複数構成 USB デバイスの USB 構成を、構成インデックス *n* に設定します。これ以降、デバイスはこの構成に設定され、この設定は、デバイスのリブート、ホットリムーブ、および構成解除/構成の後も保持されます。

n の有効な値の範囲は、`0` から (`Nconfigs -1`) までです。`disconnect` に続けて `configure` を実行すると、デバイスはリセットされます。`configure` により、デバイスは新しい構成設定に構成されます。

いずれかの手順が失敗した場合、構成ファイルとデバイスが以前の状態に復元され、エラーメッセージが発行されます。

`usb_reset`

デバイスのソフトウェアリセット (再列挙) を実行します。これは、デバイスを取り外して再度挿入するのと同等の意味を持ちます。ハブで個々のポートの電源再投入がサポートされている場合、ハブのポートで電源が一度切断され、再度投入されます。

接続されているデバイスがハブである場合、この機能には、そのハブと、そのハブをルートとするツリーの下位のデバイスをリセットするという効果があります。

いずれかの手順が失敗した場合、デバイスが以前の状態に復元され、エラーメッセージが発行されます。

状態の表: 接続点の状態とコマンド:

```
Valid states:
    empty/unconfigured         → no device connected

    disconnected/unconfigured  → logically disconnected,
                                 unavailable,
                                 devinfo node removed,
                                 device physically connected

    connected/unconfigured     → logically connected,
                                 unavailable,
                                 devinfo node present

    connected/configured       → connected, available
```

次の表は、操作またはコマンドの実行による状態の移行を示しています。

```
current state      operation           new state
-------------      ---------           ---------
empty/
unconfigured:
              device plugged in:     connected/configured or
                                     connected/unconfigured
                                     (if enumeration failed)
```

```
              device removed:        n/a
              cfgadm -c unconfigure: empty/unconfigured
              cfgadm -c configure:   empty/unconfigured
              cfgadm -c disconnect:  empty/unconfigured
                                     (no-op and error)

disconnected/
unconfigured:
              device plugged in:     n/a
              device removed:        empty/unconfigured
              cfgadm -c unconfigure: disconnected/unconfigured
              cfgadm -c configure:   connected/configured, or
                                     connected/unconfigured
                                     (if reenumeration failed)
             cfgadm -c disconnect:   disconnected/unconfigured

connected/unconfigured:
             device plugged in:      n/a
             device removed:         empty/unconfigured
             cfgadm -c unconfigure:  connected/unconfigured
             cfgadm -c configure:    connected/configured, or
                                     connected/unconfigured
                                     (if reenumeration failed)
             cfgadm -c disconnect:   disconnected/unconfigured

connected/configured:
             device plugged in:      n/a
             device removed:         empty/unconfigured or
                                     connected/configured,
                                     but with ap condition
                                     'unusable' if device
                                     was open when removed
             cfgadm -c unconfigure:  connected/unconfigured
             cfgadm -c configure:    connected/configured
             cfgadm -c disconnect:   disconnected/unconfigured
```

使用例

例 1　すべての USB デバイスのステータスの一覧表示

次のコマンドは、システム上のすべての USB デバイスのステータスを一覧表示します。

```
# cfgadm
Ap_Id           Type         Receptacle   Occupant     Condition
usb0/1          USB-hub      connected    configured   ok
usb0/2          unknown      empty        unconfigured ok
usb0/1.1        USB-storage  connected    configured   ok
usb0/1.2        unknown      empty        unconfigured ok
usb0/1.3        unknown      empty        unconfigured ok
usb0/1.4        USB-device connected    configured   ok
```

例 1 すべての USB デバイスのステータスの一覧表示 *(続き)*

cfgadm では、現時点では個々のインタフェースを制御できないため、ap_id usb0/1.4 の USB-device デバイスが単一のユニットとして処理されています。

例 2 デバイスが接続されていないポートのステータスの一覧表示

次のコマンドは、デバイスが接続されていないポートのステータスを一覧表示します。

```
example# cfgadm -l usb0/1.3
Ap_Id           Type         Receptacle   Occupant     Condition
usb0/1.3        unknown      empty        unconfigured ok
```

例 3 デバイスが接続されている同じポートのステータスの一覧表示

次のコマンドは、デバイスが物理的に接続され、問題なく構成されたあとの、同じポートのステータスを一覧表示します。

```
example# cfgadm -l usb0/1.3
Ap_Id           Type         Receptacle   Occupant     Condition
usb0/1.3        USB-hub      connected    configured   ok
```

例 4 既存の USB デバイスの構成解除

次のコマンドは、usb0/1.3 に接続された USB デバイスを構成解除し、ap_id のステータスを表示します。

```
example# cfgadm -c unconfigure usb0/1.3
Unconfigure the device: /devices/pci@0,0/pci8086,7112@7,2/hub@2:2.3
This operation suspends activity on the USB device
Continue (yes/no)?

Enter:

y

example# cfgadm -l usb0/1.3
Ap_Id           Type         Receptacle   Occupant     Condition
usb0/1.3        unknown      connected    unconfigured ok
```

例 5 既存の USB デバイスの構成解除と論理的な切断

次のコマンドは、usb0/1.3 に接続された USB デバイスを構成解除し、論理的に切断します。

```
example# cfgadm -c disconnect usb0/1.3
Disconnect the device: /devices/pci@0,0/pci8086,7112@7,2/hub@2:2.3
This operation suspends activity on the USB device
Continue (yes/no)?
```

例 5 既存の USB デバイスの構成解除と論理的な切断 *(続き)*

```
Enter:

y

example# cfgadm -l usb0/1.3
Ap_Id          Type         Receptacle     Occupant        Condition
usb0/1.3       unknown      disconnected   unconfigured    ok
```

disconnect の実行は、cfgadm によって最初に unconfigure が実行されることを暗黙的に意味します。デバイスがまだ物理的に接続されていても、受容体のステータスは disconnected と示されています。この場合、物理的なホットプラグ接続を行うか、または ap_id で cfgadm -c configure を使用すると、オンラインに戻ります。

例 6 以前に構成解除した USB デバイスの構成

次のコマンドは、以前に usb0/1.3 に接続されていた USB デバイスを構成します。

```
example # cfgadm -yc configure usb0/1.3
example# cfgadm -l usb0/1.3
Ap_Id            Type         Receptacle   Occupant     Condition
usb0/1.3         unknown      connected    configured   ok
```

例 7 USB デバイスのリセット

次のコマンドは、USB デバイスをリセットします。

```
example# cfgadm -x usb_reset usb0/1.3
Reset the device: /devices/pci@0,0/pci8086,7112@7,2/hub@2:2.3
This operation suspends activity on the USB device
Continue (yes/no)?

Enter:

y
```

例 8 単一の USB デバイスに関する詳細情報の表示

次のコマンドは、単一の USB デバイスに関する詳細情報を表示します。このデバイスでは、次の USB 固有の情報が「Information」フィールドに示されています。

- 製造元文字列: Iomega
- 製品文字列: USB Zip 250
- サポートされる構成数: 1
- 現在アクティブな構成: 0
- 構成 0 の構成文字列記述子: Default

```
example# cfgadm -lv  usb0/1.5
Ap_Id                  Receptacle   Occupant     Condition  Information
When          Type          Busy          Phys_Id
```

例 8 単一の USB デバイスに関する詳細情報の表示 *(続き)*

```
usb0/1.5     connected    configured   ok         Mfg:"Io
mega"  Product:"USB Zip 250"  NConfigs:1  Config:0 : Default
```

```
example# cfgadm -l -s "cols=ap_id:info" usb0/1.5
Ap_Id                         Information
usb0/1.5                      Mfg:"Iomega"  Product:"USB Zip 250"
NConfigs:1  Config:0 : Default
```

例 9 すべての USB デバイスに関する詳細情報の表示

次のコマンドは、システム上のすべての USB デバイスに関する詳細情報を表示します。

```
example# cfgadm -l -s "select=class(usb),cols=ap_id:info"
Ap_Id                         Information
usb0/1                        Mfg:<undefined>  Product:<undefined>
NConfigs:1  Config:0 <no cfg str descr>
usb0/2
usb0/1.1                      Mfg:<undefined>  Product:<undefined>
NConfigs:1  Config:0 <no cfg str descr>
usb0/1.2
usb0/1.3
usb0/1.4                      Mfg:"Wizard"  Product:"Modem/ISDN"
NConfigs:3  Config:1 : V.90 Analog Modem
usb0/1.5                      Mfg:"Iomega"  Product:"USB Zip 250"
NConfigs:1  Config:0 : Default
usb0/1.6                      Mfg:"SOLID YEAR"  Product:"SOLID YEAR
USB"NConfigs:1  Config:0 <no cfg str descr>
usb0/1.7
```

`ap_id` のみが含まれている行は、空のポートです。これらは、フィルタ処理によって除外できます。次の例では、接続されたデバイスがある USB `ap_id` と、それらのデバイスの情報のみが示されます。

```
example# cfgadm -l -s "select=class(usb),cols=ap_id:info" | grep Mfg
usb0/1                        Mfg:<undefined>  Product:<undefined>
NConfigs:1  Config:0 <no cfg str descr>
usb0/1.1                      Mfg:<undefined>  Product:<undefined>
NConfigs:1  Config:0 <no cfg str descr>
usb0/1.4                      Mfg:"Wizard"  Product:"Modem/ISDN"
NConfigs:3  Config:1 : V.90 Analog Modem
usb0/1.5                      Mfg:"Iomega"  Product:"USB Zip 250"
NConfigs:1  Config:0 : Default
usb0/1.6                      Mfg:"SOLID YEAR"  Product:"SOLID YEAR USB"
Config:0 <no cfg str descr>
```

例 10　複数構成 USB デバイスに関する情報の一覧表示

次の例では、複数構成 USB デバイスに関する情報が一覧表示されます。

NConfigs フィールドに注目してください。このデバイスで使用できる構成数は、0、1、および 2 (0 - (*N*Configs-1)) です。

```
example# cfgadm -l -s "cols=ap_id:info" usb0/1.4
Ap_Id                          Information
usb0/1.4                       Mfg:"Wizard"  Product:"Modem/ISDN"
NConfigs:3  Config:1 V.90 Analog Modem"
```

例 11　複数構成 USB デバイスの現在の構成の設定

次の例では、複数構成 USB デバイスの現在の構成が設定されます。

```
example# cfgadm -o config=2 -x usb_config usb0/1.4
Setting the device: /devices/pci@1f,2000/usb@1/device@3
to USB configuration 2
This operation suspends activity on the USB device
Continue (yes/no)?

Enter:

y

USB configuration changed successfully.
```

まったく同じタイプの複数のデバイスが同じバス上にある場合は、デバイスのパスを確認して、デバイスの正しいインスタンスが参照されていることを確認するようにしてください。この情報は、「Information」フィールドに示されています。

ファイル

/usr/lib/cfgadm/usb.so.1
　汎用的な USB デバイス管理のためのハードウェア固有ライブラリ

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library

関連項目

cfgadm(1M), config_admin(3CFGADM), attributes(5), scsa2usb(7D), usba(7D)

USB 1.1 の仕様 (www.usb.org)

『Oracle Solaris の管理: 一般的なタスク』

注意事項

cfgadm(1M) では、現在アプリケーションによって開かれている USB デバイスの構成解除、切断、リセット、または構成の変更を行うことはできません。また、ハブの階層内のデバイスがアプリケーションによって開かれている場合、そのハブでこれらの操作を実行すると失敗します。現在使用中の USB 外部ストレージデバイスの構成解除については、scsa2usb(7D) を参照してください。

スーパーユーザーのみが、接続点で機能を実行できます。ただし、接続点の一覧表示については、スーパーユーザーである必要はありません。

名前

chat – automated conversational exchange tool

形式

`chat [options] script`

機能説明

The `chat` program implements a conversational text-based exchange between the computer and any serial device, including (but not limited to) a modem, an ISDN TA, and the remote peer itself, establishing a connection between the Point-To-Point Protocol daemon (`pppd`) and the remote `pppd` process.

オプション

The `chat` command supports the following options:

`-f` <*chat file*>
: Read the `chat` script from the `chat` file. This option is mutually exclusive with the `chat` script parameters. You must have `read` access to use the file. Multiple lines are permitted in the file. Use the space or horizontal tab characters to separate the strings.

`-t` <*timeout*>
: Set the timeout for the expected string to be received. If the string is not received within the time limit, the reply string is not sent. If specified, a 'subexpect' (alternate reply) string can be sent. Otherwise, if no alternate reply strings remain, the `chat` script fails.. A failed script will cause the `chat` program to terminate with a non-zero error code.

`-r` <*report file*>
: Set the file for output of the report strings. If you use the keyword `REPORT`, the resulting strings are written to this file. If the `-r` option is not used and you use the `REPORT` keyword, the `stderr` file is used for the report strings.

`-e`
: Start with the echo option turned on. You turn echo on or off at specific points in the `chat` script using the `ECHO` keyword. When echoing is enabled, all output from the modem is echoed to `stderr`.

`-E`
: Enables environment variable substitution within `chat` scripts using the standard *$xxx* syntax.

`-v`
: Request that the `chat` script execute in a verbose mode. The `chat` program logs the execution state of the `chat` script as well as all text received from the modem and output strings sent to the modem. The default is to log through `syslog(3C)` with facility `local2`; the logging method is alterable using the `-S` and `-s` options.

`-V`
: Request that the `chat` script be executed in a `stderr` verbose mode. The `chat` program logs all text received from the modem and output strings sent to the modem to `stderr`. `stderr` is usually the local console at the station running the `chat` or `pppd` program.

`-s`
: Use `stderr`. Log messages from `-v` and error messages are sent to `stderr`.

`-S`	Do not use syslog. By default, error messages are set to syslog. This option prevents log messages from `-v` and error messages from being sent to syslog.
`-T` <*phone number*>	Pass in an arbitrary string (usually a telephone number) that will be substituted for the `\T` substitution metacharacter in a send string.
`-U` <*phone number 2*>	Pass in a second string (usually a telephone number) that will be substituted for the `\U` substitution metacharacter in a send string. This is useful when dialing an ISDN terminal adapter that requires two numbers.
`script`	If the script is not specified in a file with the `-f` option, the script is included as parameters to the `chat` program.

拡張機能説明

Chat Script

The `chat` script defines communications. A script consists of one or more "expect-send" pairs of strings separated by spaces, with an optional "subexpect-subsend" string pair, separated by a dash (as in the following example:)

ogin:-BREAK-ogin: ppp ssword: hello2u2

The example indicates that the `chat` program should expect the string "ogin:". If it fails to receive a login prompt within the time interval allotted, it sends a break sequence to the remote and then expects the string "ogin:". If the first "ogin:" is received, the break sequence is not generated.

Upon receiving the login prompt, the `chat` program sends the string ”ppp” and then expects the prompt "ssword:". When the password prompt is received, it sends the password hello2u2.

A carriage return is normally sent following the reply string. It is not expected in the "expect" string unless it is specifically requested by using the `\r` character sequence.

The expect sequence should contain only what is needed to identify the received data. Because it's stored on a disk file, it should not contain variable information. Generally it is not acceptable to look for time strings, network identification strings, or other variable pieces of data as an expect string.

To correct for characters that are corrupted during the initial sequence, look for the string "ogin:" rather than "login:". The leading "l" character may be received in error, creating problems in finding the string. For this reason, scripts look for "ogin:" rather than "login:" and "ssword:" rather than "password:".

An example of a simple script follows:

```
ogin: ppp ssword: hello2u2
```

The example can be intrepreted as: expect ogin:, send ppp, expect ...ssword:, send hello2u2.

When login to a remote peer is necessary, simple scripts are rare. At minimum, you should include sub-expect sequences in case the original string is not received. For example, consider the following script:

```
ogin:--ogin: ppp ssword: hello2u2
```

This script is more effective than the simple one used earlier. The string looks for the same login prompt; however, if one is not received, a single return sequence is sent and then the script looks for login: again. If line noise obscures the first login prompt, send the empty line to generate a login prompt again.

Comments

Comments can be embedded in the `chat` script. Comment lines are ignored by the chat program. A comment starts with the hash (“#”) character in column one. If a # character is expected as the first character of the expect sequence, quote the expect string. If you want to wait for a prompt that starts with a # character, write something like this:

```
# Now wait for the prompt and send logout string
’# ’ logout
```

Sending Data From A File

If the string to send begins with an at sign (“@”), the remainder of the string is interpreted as the name of the file that contains the string. If the last character of the data read is a newline, it is removed. The file can be a named pipe (or fifo) instead of a regular file. This enables `chat` to communicate with another program, for example, a program to prompt the user and receive a password typed in.

Abort

Many modems report the status of a call as a string. These status strings are often “CONNECTED” or "NO CARRIER" or "BUSY." If the modem fails to connect to the remote, you can terminate the script. Abort strings may be specified in the script using the ABORT sequence. For example:

```
ABORT BUSY ABORT ’NO CARRIER’ ’’ ATZ OK ATDT5551212 CONNECT
```

This sequence expects nothing and sends the string ATZ. The expected response is the string OK. When OK is received, the string ATDT5551212 dials the telephone. The expected string is CONNECT. If CONNECT is received, the remainder of the script is executed. When the modem finds a busy telephone, it sends the string BUSY, causing the string to match the abort character sequence. The script fails because it found a match to the abort string. If the NO CARRIER string is received, it aborts for the same reason.

Clr_Abort

The CLR_ABORT sequence clears previously set ABORT strings. ABORT strings are kept in an array of a pre-determined size; CLR_ABORT reclaims the space for cleared entries, enabling new strings to use that space.

Say

The SAY string enables the script to send strings to a user at a terminal via standard error. If `chat` is being run by `pppd` and `pppd` is running as a daemon (detached from its controlling terminal), standard error is normally redirected to the `/etc/ppp/connect-errors` file.

SAY strings must be enclosed in single or double quotes. If carriage return and line feed are required for the output, you must explicitly add them to your string.

The SAY string can provide progress messages to users even with "ECHO OFF." For example, add a line similar to the following to the script:

```
ABORT BUSY
ECHO OFF
SAY "Dialing your ISP...\n"
'' ATDT5551212
TIMEOUT 120
SAY "Waiting up to 2 minutes for connection ..."
CONNECT ''
SAY "Connected, now logging in ...\n"
ogin: account
ssword: pass
$ \c
SAY "Logged in OK ... \n"
```

This sequence hides script detail while presenting the SAY string to the user. In this case, you will see:

```
Dialing your ISP...
Waiting up to 2 minutes for connection...Connected, now logging in...
Logged in OK ...
```

Report

REPORT is similar to the ABORT string. With REPORT, however, strings and all characters to the next control character (such as a carriage return), are written to the report file.

REPORT strings can be used to isolate a modem's transmission rate from its CONNECT string and return the value to the `chat` user. Analysis of the REPORT string logic occurs in conjunction with other string processing, such as looking for the expect string. It's possible to use the same string for a REPORT and ABORT sequence, but probably not useful.

Report strings may be specified in the script using the REPORT sequence. For example:

```
REPORT CONNECT
ABORT BUSY
ATDT5551212 CONNECT
ogin: account
```

The above sequence expects nothing, then sends the string ATDT5551212 to dial the telephone. The expected string is CONNECT. If CONNECT is received, the remainder of the

script is executed. In addition, the program writes the string CONNECT to the report file (specified by `-r`) in addition to any characters that follow.

Clr_Report

CLR_REPORT clears previously set REPORT strings. REPORT strings are kept in an array of a pre-determined size; CLR_REPORT reclaims the space for cleared entries so that new strings can use that space.

Echo

ECHO determines if modem output is echoed to `stderr`. This option may be set with the `-e` option, but can also be controlled by the ECHO keyword. The "expect-send" pair ECHO ON enables echoing, and ECHO OFF disables it. With ECHO, you can select which parts of the conversation should be visible. In the following script:

```
ABORT   'BUSY'
ABORT   'NO CARRIER'
""    AT&F
OK\r\n  ATD1234567
\r\n    \c
ECHO    ON
CONNECT \c
ogin:   account
```

All output resulting from modem configuration and dialing is not visible, but output is echoed beginning with the CONNECT (or BUSY) message.

Hangup

The HANGUP option determines if a modem hangup is considered as an error. HANGUP is useful for dialing systems that hang up and call your system back. HANGUP can be ON or OFF. When HANGUP is set to OFF and the modem hangs up (for example, following the first stage of logging in to a callback system), `chat` continues running the script (for example, waiting for the incoming call and second stage login prompt). When the incoming call is connected, use the HANGUP ON string to reinstall normal hang up signal behavior. An example of a simple script follows:

```
ABORT   'BUSY'
 ""  AT&F
 OK\r\n  ATD1234567
 \r\n    \c
 CONNECT \c
 'Callback login:' call_back_ID
 HANGUP OFF
 ABORT "Bad Login"
 'Callback Password:' Call_back_password
 TIMEOUT 120
 CONNECT \c
 HANGUP ON
 ABORT "NO CARRIER"
 ogin:--BREAK--ogin: real_account
```

Timeout

The initial timeout value is 45 seconds. Use the `-t` parameter to change the intial timeout value.

To change the timeout value for the next expect string, the following example can be used:

```
''"AT&F
 OK ATDT5551212
 CONNECT \c
 TIMEOUT 10
 ogin:--ogin: username
 TIMEOUT 5
 assword: hello2u2
```

The example changes the timeout to ten seconds when it expects the login: prompt. The timeout is changed to five seconds when it looks for the password prompt.

Once changed, the timeout value remains in effect until it is changed again.

EOT

The EOT special reply string instructs the `chat` program to send an EOT character to the remote. This is equivalent to using ^D\c as the reply string. The EOT string normally indicates the end-of-file character sequence. A return character is not sent following the EOT. The EOT sequence can embedded into the send string using the sequence ^D.

BREAK

The BREAK special reply string sends a break condition. The break is a special transmitter signal. Many UNIX systems handle break by cycling through available bit rates, and sending break is often needed when the remote system does not support autobaud. BREAK is equivalent to using \K\c as the reply string. You embed the break sequence into the send string using the \K sequence.

Escape Sequences

Expect and reply strings can contain escape sequences. Reply strings accept all escape sequences, while expect strings accept most sequences. A list of escape sequences is presented below. Sequences that are not accepted by expect strings are indicated.

`''` Expects or sends a null string. If you send a null string, `chat` sends the return character. If you expect a null string, `chat` proceeds to the reply string without waiting. This sequence can be a pair of apostrophes or quote mark characters.

`\b` Represents a backspace character.

`\c` Suppresses the newline at the end of the reply string. This is the only method to send a string without a trailing return character. This sequence must be at the end of the send string. For example, the sequence hello\c will simply send the characters h, e, l, l, o. (Not valid in expect.)

`\d` Delay for one second. The program uses `sleep(1)` which delays to a maximum of one second. (Not valid in expect.)

`\K` Insert a BREAK. (Not valid in expect.)

`\n` Send a newline or linefeed character.

`\N` Send a null character. The same sequence may be represented by \0. (Not valid in expect.)

`\p` Pause for 1/10th of a second. (Not valid in expect.)

`\q` Suppress writing the string to syslog. The string ?????? is written to the log in its place. (Not valid in expect.)

`\r` Send or expect a carriage return.

`\s` Represents a space character in the string. Can be used when it is not desirable to quote the strings which contains spaces. The sequence 'HI TIM' and HI\sTIM are the same.

`\t` Send or expect a tab character.

`\T` Send the phone number string as specified with the `-T` option. (Not valid in expect.)

`\U` Send the phone number 2 string as specified with the `-U` option. (Not valid in expect.)

`\\` Send or expect a backslash character.

`\ddd` Collapse the octal digits (ddd) into a single ASCII character and send that character. (\000 is not valid in an expect string.)

`^C` Substitute the sequence with the control character represented by C. For example, the character DC1 (17) is shown as ^Q. (Some characters are not valid in expect.)

環境変数

Environment variables are available within `chat` scripts if the `-E` option is specified on the command line. The metacharacter `$` introduces the name of the environment variable to substitute. If the substition fails because the requested environment variable is not set, nothing is replaced for the variable.

終了ステータス

The `chat` program terminates with the following completion codes:

`0` Normal program termination. Indicates that the script was executed without error to normal conclusion.

`1` One or more of the parameters are invalid or an expect string was too large for the internal buffers. Indicates that the program was not properly executed.

`2` An error occurred during the execution of the program. This may be due to a read or write operation failing or `chat` receiving a signal such as SIGINT.

`3` A timeout event occurred when there was an expect string without having a "-subsend" string. This indicates that you may not have programmed the script correctly for the condition or that an unexpected event occurred and the expected string could not be found.

4 The first string marked as an ABORT condition occurred.

5 The second string marked as an ABORT condition occurred.

6 The third string marked as an ABORT condition occurred.

7 The fourth string marked as an ABORT condition occurred.

. . . The other termination codes are also strings marked as an ABORT condition.

To determine which event terminated the script, use the termination code. It is possible to decide if the string "BUSY" was received from the modem versus "NO DIALTONE." While the first event may be retried, the second probably will not succeed during a retry.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ppp
Interface Stability	Committed

関連項目

sleep(1), uucp(1C), pppd(1M), uucico(1M), syslog(3C), attributes(5)

Additional information on chat scripts are available with UUCP documentation. The chat script format was taken from scripts used by the uucico program.

名前

check-hostname – check if sendmail can determine the system's fully-qualified host name

形式

```
/usr/sbin/check-hostname
```

機能説明

The `check-hostname` script is a migration aid for `sendmail(1M)`. This script tries to determine the local host's fully-qualified host name (FQHN) in a manner similar to `sendmail(1M)`. If `check-hostname` is able to determine the FQHN of the local host, it reports success. Otherwise, `check-hostname` reports how to reconfigure the system so that the FQHN can be properly determined.

ファイル

`/etc/hosts`	Host name database
`/etc/nsswitch.conf`	Name service switch configuration file
`/etc/resolv.conf`	Configuration file for name server routines

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail
Interface Stability	Committed

関連項目

`domainname(1M)`, `sendmail(1M)`, `hosts(4)`, `attributes(5)`

名前

check-permissions – check permissions on mail rerouting files

形式

`/usr/sbin/check-permissions [login]`

機能説明

The `check-permissions` script is intended as a migration aid for `sendmail(1M)`. It checks the `/etc/mail/sendmail.cf` file for all configured alias files, and checks the alias files for `:include:` files. It also checks for certain `.forward` files. For each file that `check-permissions` checks, it verifies that none of the parent directories are group- or world-writable. If any directories are overly permissive, it is reported. Otherwise it reports that no unsafe directories were found.

As to which `.forward` files are checked, it depends on the arguments included on the command line. If no argument is given, the current user's home directory is checked for the presence of a `.forward` file. If any arguments are given, they are assumed to be valid logins, and the home directory of each one is checked.

If the special argument `ALL` is given, the `passwd` entry in the `/etc/nsswitch.conf` file is checked, and all password entries that can be obtained through the switch file are checked. In large domains, this can be time-consuming.

オペランド

The following operands are supported:

login Where *login* is a valid user name, checks the home directory for *login*.

`ALL` Checks the home directory of *all* users.

ファイル

`/etc/mail/sendmail.cf` Defines environment for `sendmail`

`/etc/mail/aliases` Ascii mail aliases file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail
Interface Stability	Committed

関連項目

`getent(1M)`, `sendmail(1M)`, `aliases(4)`, `attributes(5)`

名前

chk_encodings – ラベルエンコーディングファイルの構文の検査

形式

`/usr/sbin/chk_encodings [-a] [-c` *maxclass*`] [`*pathname*`]`

機能説明

`chk_encodings` は、*pathname* で指定されたラベルエンコーディングファイルの構文を検査します。`-a` オプションを使用すると、`chk_encodings` は、*pathname* で指定されたラベルエンコーディングファイルの意味解析も出力します。*pathname* が指定されない場合、`chk_encodings` は `/etc/security/tsol/label_encodings` ファイルを検査および解析します。

ラベルエンコーディングファイルの解析が要求された場合、エラーが検出された場合でも、実施可能な解析が標準出力ファイルに書き込まれます。

オプション

`-a` ラベルエンコーディングファイルの意味解析を実行します。

`-c` *maxclass* ラベルエンコーディングファイルの `CLASSIFICATIONS` セクションに指定可能な最大格付け値 *maxclass* (デフォルトは 255) を指定します。

終了ステータス

正常終了時、`chk_encodings` は終了ステータス `0` (真) を返し、*pathname* 内にエラーが検出されなかったことを示すメッセージを標準出力ファイルに書き込みます。それ以外の場合、`chk_encodings` はゼロ以外 (偽) の終了コードを返し、診断メッセージを標準出力ファイルに書き込みます。

ファイル

`/etc/security/tsol/label_encodings`
ラベルエンコーディングファイルには、このシステムの定義済みラベルに対する格付け名、語句、制約、および値が含まれます。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。
標準	DDS-2600-6216-93、『『Compartmented Mode Workstation Labeling: Encodings Format』』、1993 年 9 月

コマンド出力は非インタフェースです。コマンド呼び出しは、DIA MAC ポリシーを実装するシステムでは「確実」です。

関連項目

`label_encodings(4)`, `attributes(5)`, `labels(5)`

『Trusted Extensions Label Administration』の「How to Analyze and Verify the label_encodings File」

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

このファイルは、米国国防情報局 (DIA) の必須アクセス制御 (MAC) ポリシーの一部です。このファイルは、Solaris Trusted Extensions ソフトウェアの将来のリリース向けに作成されるその他の MAC ポリシーには適用されない可能性があります。

名前

chroot – change root directory for a command

形式

`/usr/sbin/chroot` *newroot command*

機能説明

The `chroot` utility causes *command* to be executed relative to *newroot*. The meaning of any initial slashes (/) in the path names is changed to *newroot* for *command* and any of its child processes. Upon execution, the initial working directory is *newroot*.

Notice that redirecting the output of *command* to a file,

`chroot` *newroot command* `>x`

will create the file x relative to the original root of *command*, not the new one.

The new root path name is always relative to the current root. Even if a `chroot` is currently in effect, the *newroot* argument is relative to the current root of the running process.

This command can be run only by the super-user.

戻り値

The exit status of `chroot` is the return value of *command*.

使用例

例 1　Using the chroot Utility

The `chroot` utility provides an easy way to extract `tar` files (see `tar(1)`) written with absolute filenames to a different location. It is necessary to copy the shared libraries used by `tar` (see `ldd(1)`) to the *newroot* filesystem.

```
example# mkdir /tmp/lib; cd /lib
example# cp ld.so.1 libc.so.1 libcmd.so.1 libdl.so.1 \
         libsec.so.1 /tmp/lib
example# cp /usr/bin/tar /tmp
example# dd if=/dev/rmt/0 | chroot /tmp tar xvf -
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`cd(1)`, `tar(1)`, `chroot(2)`, `ttyname(3C)`, `attributes(5)`

注意事項

Exercise extreme caution when referencing device files in the new root file system.

References by routines such as `ttyname(3C)` to stdin, stdout, and stderr will find that the device associated with the file descriptor is unknown after `chroot` is run.

名前

cimworkshop – start the Sun WBEM CIM WorkShop application

形式

`/usr/sadm/bin/cimworkshop`

機能説明

The `cimworkshop` command starts Sun WBEM CIM WorkShop, a graphical user interface that enables you to create, modify, and view the classes and instances that describe the managed resources on your system.

Managed resources are described using a standard information model called Common Information Model (CIM). A CIM class is a computer representation, or model, of a type of managed resource, such as a printer, disk drive, or CPU. A CIM instance is a particular managed resource that belongs to a particular class. Instances contain actual data. Objects can be shared by any WBEM-enabled system, device, or application. CIM objects are grouped into meaningful collections called schema. One or more schemas can be stored in directory-like structures called namespaces.

The CIM WorkShop application displays a Login dialog box. Context help is displayed on the left side of the CIM WorkShop dialog boxes. When you click on a field, the help content changes to describe the selected field.

By default, CIM WorkShop uses the RMI protocol to connect to the CIM Object Manager on the local host, in the default namespace, `root\cimv2`. You can select HTTP if you want to communicate to a CIM Object Manager using the standard XML/HTTP protocol from the Desktop Management Task Force. When a connection is established, all classes contained in the default namespace are displayed in the left side of the CIM WorkShop window.

The name of the current namespace is listed in the tool bar. All programming operations are performed within a namespace. Four namespaces are created in a `root` namespace during installation:

cimv2	Contains the default CIM classes that represent managed resources on your system.
security	Contains the security classes used by the CIM Object Manager to represent access rights for users and namespaces.
system	Contains properties for configuring the CIM Object Manager.
snmp	Contains pre-defined SNMP-related classes and all SNMP MOF files that are compiled.

The `cimworkshop` application allows you to perform the following tasks:

Create, view, and change namespaces.	Use the CIM WorkShop application to view all namespaces. A namespace is a directory-like structure that can store CIM classes and instances.

Create, delete, and view CIM classes.	You cannot modify the unique attributes of the classes that make up the CIM and Solaris Schema. You can create a new instance or subclass of the class and modify the desired attributes in that instance or subclass.
Create, modify, delete, and view CIM instances.	You can add instances to a class and modify its inherited properties or create new properties. You can also change the property values of a CIM instance.
Invoke methods.	You can set input values for a parameter of a method and invoke the method.

When CIM WorkShop connects to the CIM Object Manager in a particular namespace, all subsequent operations occur within that namespace. When you connect to a namespace, you can access the classes and instances in that namespace (if they exist) and in any namespaces contained in that namespace.

When you use CIM WorkShop to view CIM data, the WBEM system validates your login information on the current host. By default, a validated WBEM user is granted read access to the CIM Schema. The CIM Schema describes managed objects on your system in a standard format that all WBEM-enabled systems and applications can interpret.

Read Only	Allows read-only access to CIM Schema objects. Users with this privilege can retrieve instances and classes, but cannot create, delete, or modify CIM objects.
Read/Write	Allows full read, write, and delete access to all CIM classes and instances.
Write	Allows write and delete, but not read access to all CIM classes and instances.
None	Allows no access to CIM classes and instances.

使用法

The `cimworkshop` command is not a tool for a distributed environment. Rather, this command is used for local administration on the machine on which the CIM Object Manager is running.

終了ステータス

The `cimworkshop` utility terminates with exit status `0`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbdev

関連項目 mofcomp(1M), wbemlogviewer(1M), init.wbem(1M), attributes(5)

名前

clear_locks – clear locks held on behalf of an NFS client

形式

`/usr/sbin/clear_locks [-s]` *hostname*

機能説明

The `clear_locks` command removes all file, record, and share locks created by the *hostname* and held on the current host, regardless of which process created or owns the locks.

This command can be run only by the super-user.

This command should only be used to repair the rare case of a client crashing and failing to clear held locks. Clearing locks held by an active client may cause applications to fail in an unexpected manner.

オプション

`-s` Remove all locks created by the current machine and held by the server *hostname*.

オペランド

The following operands are supported:

hostname name of host server

終了ステータス

`0` Successful operation.

`1` If not root.

`2` Usage error.

`3` If unable to contact server (RPC).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fcntl(2)`, `attributes(5)`

名前

clinfo – display cluster information

形式

```
clinfo [-nh]
```

機能説明

The `clinfo` command displays cluster configuration information about the node from which the command is executed.

Without arguments, `clinfo` returns an exit status of `0` if the node is configured and booted as part of a cluster. Otherwise, `clinfo` returns an exit status of `1`.

オプション

The following options are supported:

`-h` Displays the highest node number allowed to be configured. This is different from the maximum number of nodes supported in a given cluster. The current highest configured node number can change immediately after the command returns since new nodes can be dynamically added to a running cluster.

For example, `clinfo -h` might return `64`, meaning that the highest number you can use to identify a node is `64`. See the Sun Cluster 3.0 System Administration Guide for a description of utilities you can use to determine the number of nodes in a cluster.

`-n` Prints the number of the node from which `clinfo` is executed.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

This is usually because the node is not configured or booted as part of a cluster.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`attributes(5)`

名前

clri, dcopy – clear inode

形式

`clri [-F` *FSType*`] [-V]` *special i-number*

`dcopy [-F` *FSType*`] [-V]` *special i-number*

機能説明

`clri` writes zeros on the inodes with the decimal *i-number* on the file system stored on *special*. After `clri`, any blocks in the affected file show up as missing in an `fsck(1M)` of *special*.

Read and write permission is required on the specified file system device. The inode becomes allocatable.

The primary purpose of this routine is to remove a file that for some reason appears in no directory. If it is used to zap an inode that does appear in a directory, care should be taken to track down the entry and remove it. Otherwise, when the inode is reallocated to some new file, the old entry will still point to that file. At that point, removing the old entry will destroy the new file. The new entry will again point to an unallocated inode, so the whole cycle is likely to be repeated again and again.

`dcopy` is a symbolic link to `clri`.

オプション

`-F` *FSType*
: Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by matching *special* with an entry in the table, or by consulting `/etc/default/fs`.

`-V`
: Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from `/etc/vfstab`. This option should be used to verify and validate the command line.

使用法

See `largefile(5)` for the description of the behavior of `clri` and `dcopy` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/default/fs` Default local file system type

`/etc/vfstab` List of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fsck(1M)`, `vfstab(4)`, `attributes(5)`, `largefile(5)`

注意事項

This command might not be supported for all *FSTypes*.

名前

consadm – select or display devices used as auxiliary console devices

形式

```
/usr/sbin/consadm

/usr/sbin/consadm [-a device. . .] [-p]

/usr/sbin/consadm [-d device. . .] [-p]

/usr/sbin/consadm [-p]
```

機能説明

`consadm` selects the hardware *device* or devices to be used as auxiliary console devices, or displays the current device. Only superusers are allowed to make or display auxiliary console device selections.

Auxiliary console devices receive copies of `console` messages, and can be used as the console during `single user` mode. In particular, they receive kernel messages and messages directed to `/dev/sysmsg`. On Solaris x86 based systems they can also be used for interaction with the bootstrap.

By default, selecting a display device to be used as an auxiliary console device selects that device for the duration the system remains up. If the administrator needs the selection to persist across reboots the `-p` option can be specified.

`consadm` runs a daemon in the background, monitoring auxiliary console devices. Any devices that are disconnected (hang up, lose carrier) are removed from the auxiliary console device list, though not from the persistent list. While auxiliary console devices may have been removed from the device list receiving copies of `console` messages, those messages will always continue to be displayed by the default console device.

The daemon will not run if it finds there are not any auxiliary devices configured to monitor. Likewise, after the last auxiliary console is removed, the daemon will shut itself down. Therefore the daemon persists for only as long as auxiliary console devices remain active.

See `eeprom`(1M) for instructions on assigning an auxiliary console device as the system console.

オプション

The following options are supported:

`-a` *device*
: Adds *device* to the list of auxiliary console devices. Specify *device* as the path name to the device or devices to be added to the auxiliary console device list.

`-d` *device*
: Removes *device* from the list of auxiliary console devices. Specify *device* as the path name to the device or devices to be removed from the auxiliary console device list.

`-p`
: Prints the list of auxiliary consoles that will be auxiliary across reboots.

  When invoked with the `-a` or `-d` options , tells the application to make the change persist across reboot.

使用例

例 1 Adding to the list of devices that will receive console messages

The following command adds /dev/term/a to the list of devices that will receive console messages.

```
example# consadm -a /dev/term/a
```

例 2 Removing from the list of devices that will receive console messages

The following command removes /dev/term/a from the list of devices that will receive console messages. This includes removal from the persistent list.

```
example# consadm -d -p /dev/term/a
```

例 3 Printing the list of devices selected as auxiliary console devices

The following command prints the name or names of the device or devices currently selected as auxiliary console devices.

```
example# consadm
```

環境変数

See environ(5) for descriptions of the following environment variables that affect the execution of consadm: LC_CTYPE, LC_MESSAGES, and NLSPATH.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

kmdb(1), svcs(1), eeprom(1M), svcadm(1M), syslogd(1M), environ(5), attributes(5), smf(5), sysmsg(7D), console(7D)

注意事項

Auxiliary console devices are not usable for kmdb or firmware I/O, do not receive panic messages, and do not receive output directed to /dev/console.

The consadm service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/consadm
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

console-reset – force console login to display

形式

`svc:/system/console-reset`

機能説明

The `console-reset` service runs if X is not running on console and the console is ready for interactive use. Under these conditions, `console-reset` turns off the boot-time graphics, forcing the console login prompt to display.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Uncommitted

関連項目

`svcs(1)`, `svcadm(1M)`, `vbiosd(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `console-reset` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/console-reset:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

conv_lp – convert LP configuration

形式

`conv_lp [-d dir] [-f file]`

機能説明

`conv_lp` reads LP printer configuration information from a directory and converts it to an output file for use with print client software.

オプション

The following options are supported:

`-d`*dir* The root (`/`) directory from which LP configuration information is read. The default is root (`/`).

`-f`*file* The output file to which `conv_lp` writes the converted LP configuration information.

終了ステータス

The following exit values are returned:

`0` Successful completion.

non-zero An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/lp/print-client-commands

関連項目

`attributes(5)`

名前

conv_lpd – convert LPD configuration

形式

`conv_lpd [-c printcap] [-n] file`

機能説明

`conv_lpd` converts LPD printer configuration information from *file* to a `printcap` file. *file* specifies the name of the input file, and is in `printcap` format.

オプション

The following options are supported:

`-c printcap` Specifies the type of output file produced by the conversion. `-c printcap` converts to a `printcap` file.

`-n` Preserves the namelist during the conversion.

オペランド

The following operands are supported:

file The file to be converted.

終了ステータス

The following exit values are returned:

`0` Successful completion.

non-zero An error occurred.

ファイル

`/etc/printcap` SunOS 4.*x* printer capability database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/lp/print-client-commands

関連項目

`lpset(1M)`, `attributes(5)`

名前

coreadm – コアファイルの管理

形式

```
coreadm [-g pattern] [-G content] [-i pattern] [-I content]
     [-d option]... [-e option]...

coreadm [-p pattern] [-P content] [pid]...
```

機能説明

coreadm コマンドを使用して、異常終了するプロセスにより生成されるコアファイルの名前と場所を指定できます。core(4) を参照してください。

「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、「形式」の最初の形式を実行できます。この形式では、大域的なコアファイル名のパターンおよび init(1M) プロセスのコアファイル名のパターンを含む、システム全体のコアファイルオプションを構成します。すべての設定が恒久的に保存され、ブート時に適用されます。

非特権ユーザーは、「形式」の 2 番目の形式を実行できます。この形式では、オペレーティングシステムがプロセスごとのコアファイルを生成するときに使用する、ファイル名のパターンとコアファイルのコンテンツを指定します。

コアファイル名のパターンは、% から始まる文字で指定される組み込み変数を含んだ、通常のファイルシステムのパス名です。この変数は、オペレーティングシステムがコアファイルを生成するときに有効な値から展開されます。使用可能な組み込み変数は次のとおりです。

%d
: 実行ファイルのディレクトリ名。最大文字数は MAXPATHLEN

%f
: 実行ファイルの名前。最大文字数は MAXCOMLEN

%g
: 実効グループ ID

%m
: マシン名 (uname -m)

%n
: システムノード名 (uname -n)

%p
: プロセス ID

%t
: time(2) の 10 進数の値

%u
: 実効ユーザー ID

%z
: プロセスが実行されているゾーン名 (zonename)

%%
リテラル %

たとえば、コマンドが foo でプロセス ID が 1234 の場合、コアファイル名のパターン /var/cores/core.%f.%p は /var/cores/core.foo.1234 になります。

コアファイルコンテンツの記述は、プロセスのバイナリイメージを識別する一連のトークンを使用して指定されます。

anon
メインスレッドスタックではないスレッドスタックを含む、匿名プライベートマッピング

ctf
ロードされたオブジェクトファイルの CTF タイプ情報セクション

data
書き込み可能プライベートファイルマッピング

dism
DISM マッピング

heap
プロセスヒープ

ism
ISM マッピング

rodata
読み取り専用プライベートファイルマッピング

shanon
匿名共有マッピング

shfile
ファイルによってバックアップされる共有マッピング

shm
System V 共有メモリー

stack
プロセススタック

symtab
ロードされたオブジェクトファイルのシンボルテーブルセクション

text
読み取り可能および実行可能なプライベートファイルマッピング

また、トークン all を使用すると、コアファイルにプロセスのバイナリイメージのすべての部分を含めるよう指定できます。トークン none を使用すると、マッピングをまったく含めないよう指定できます。default トークンを使用すると、システ

ムのデフォルトコンテンツ
(stack+heap+shm+ism+dism+text+data+rodata+anon+shanon+ctf+symtab) を含めるよう指定できます。/proc ファイルシステムデータ構造は、マッピングコンテンツに関係なく、常にコアファイル内に存在します。

+ および - を使用して、トークンを連結できます。たとえば、コアファイルコンテンツ default-ism は、詳細共有メモリーマッピングを除くマッピングのデフォルトセットを使用してコアファイルを生成します。

引数なしの coreadm コマンドは、現在のシステム構成を報告します。たとえば、次のようになります。

```
$ coreadm
    global core file pattern: /var/cores/core.%f.%p
    global core file content: all
      init core file pattern: core
      init core file content: default
           global core dumps: enabled
      per-process core dumps: enabled
     global setid core dumps: enabled
per-process setid core dumps: disabled
    global core dump logging: disabled
```

プロセス ID のリストのみを指定した coreadm コマンドは、各プロセスにおけるプロセスごとのコアファイル名のパターンを報告します。たとえば、次のようになります。

```
$ coreadm 278 5678
  278:   core.%f.%p default
  5678:  /home/george/cores/%f.%p.%t all-ism
```

プロセスの所有者または proc_owner 特権を持つユーザーのみがこの方法でプロセスを調べることができます。

プロセスがコアをダンプしているとき、次のように最大で 3 つのコアファイルを生成できます。1 つはプロセスごとの場所、1 つはシステム全体の大域的な場所、さらに、プロセスが局所 (非大域) ゾーンで実行されていた場合は、1 つをそのプロセスが実行されていたゾーンの大域的な場所に生成できます。各コアファイルは、対応する場所に有効なオプションに従って生成されます。

生成時に、大域コアファイルはモード 600 で作成され、スーパーユーザーによって所有されます。非特権ユーザーはそのようなファイルを調べることができません。

通常のプロセスごとのコアファイルは、プロセスの資格に基づいてモード 600 で作成されます。プロセスの所有者はそのようなファイルを調べることができます。

setuid か setgid であるプロセス、またはこのプロセスの最後の exec(2) 以降に setuid か setgid だったことのあるプロセスには、コアのダンプに関連するセキュリティーの問題が存在します。同様に、最初にスーパーユーザー特権を持ってい

て、setuid(2) によってそれらの特権を失ったプロセスにも、コアのダンプに関連するセキュリティーの問題が存在します。どちらのタイプのプロセスにも、そのアドレス空間の中に、現在の非特権プロセス所有者がアクセスできるべきではない機密情報が含まれている可能性があります。setid コアファイルが有効な場合、それらはモード 600 で作成され、スーパーユーザーによって所有されます。

オプション

次のオプションがサポートされています。

-d *option...*
指定したコアファイルオプションを使用不可にします。指定可能なオプションについては、-e *option* を参照してください。

複数の -e および -d オプションをコマンド行で指定できます。「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-e *option...*
指定したコアファイルオプションを使用可能にします。*option* には次のいずれかを指定します。

大域 (global)
大域コアパターンを使用するコアダンプを許可します。

global-setid
大域コアパターンを使用する set-id コアダンプを許可します。

log
大域コアファイルの生成が試行されるときに syslog(3C) メッセージを生成します。

process
プロセスごとのコアパターンを使用するコアダンプを許可します。

proc-setid
プロセスごとのコアパターンを使用する set-id コアダンプを許可します。

複数の -e および -d オプションをコマンド行で指定できます。「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-g *pattern*
大域コアファイル名のパターンを *pattern* に設定します。パターンは / から始める必要があり、「機能説明」で説明されている特殊な % 変数のいずれも含めることができます。

「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-G *content*
大域コアファイルコンテンツを content に設定します。コンテンツは、「機能説明」で説明されているトークンを使用して指定する必要があります。

「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-i *pattern*
デフォルトのプロセスごとのコアファイル名を *pattern* に設定します。プロセスごとのパターンがまだデフォルトに設定されているプロセスは、これによってプロセスごとのパターンが変更されます。プロセスごとのパターンが設定されているプロセス、またはプロセスごとのパターンが設定されているプロセスの子孫のプロセス (-p オプションを使用) は、影響を受けません。このデフォルトは、リブート後も持続します。

「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-I *content*
デフォルトのプロセスごとのコアファイルコンテンツを *content* に設定します。プロセスごとのコンテンツがまだデフォルトに設定されているプロセスは、これによってプロセスごとのコンテンツが変更されます。プロセスごとのコンテンツが設定されているプロセス、またはプロセスごとのコンテンツが設定されているプロセスの子孫のプロセス (-P オプションを使用) は、影響を受けません。このデフォルトは、リブート後も持続します。

「Maintenance and Repair」RBAC プロファイルに属しているユーザーおよび役割だけが、このオプションを使用できます。

-p *pattern*
指定した各プロセス ID の、プロセスごとのコアファイル名のパターンを *pattern* に設定します。パターンには、「機能説明」で説明されている特殊な % 変数を含めることができます。/ から始める必要はありません。パターンが / から始まっていない場合は、プロセスがコアファイルを生成したときのカレントディレクトリから相対的に判断されます。

非特権ユーザーは、そのユーザーが所有するプロセスにのみ -p オプションを適用できます。proc_owner 特権を持つユーザーは、任意のプロセスにこのオプションを適用できます。プロセスごとのコアファイル名のパターンは、影響を受けるプロセスの将来の子プロセスに継承されます。fork(2) を参照してください。

プロセス ID が指定されていない場合、-p オプションは、親プロセス (通常、coreadm を実行したシェル) についてプロセスごとのコアファイル名のパターンを *pattern* に設定します。

-P *content*
指定した各プロセス ID のプロセスごとのコアファイルコンテンツを *content* に設定します。コンテンツは、「機能説明」で説明されているトークンを使用して指定する必要があります。

非特権ユーザーは、そのユーザーが所有するプロセスにのみ -p オプションを適用できます。proc_owner 特権を持つユーザーは、任意のプロセスにこのオプションを適用できます。プロセスごとのコアファイル名のパターンは、影響を受けるプロセスの将来の子プロセスに継承されます。fork(2) を参照してください。

プロセス ID が指定されていない場合、-P オプションは、親プロセス (通常、coreadm を実行したシェル) についてプロセスごとのファイルコンテンツを *content* に設定します。

オペランド

次のオペランドがサポートされています。

pid
プロセス ID

使用例

例 1 コアファイル名パターンの設定

ユーザーの $HOME/.profile または $HOME/.login から実行した場合、次のコマンドはログインセッション中に実行されるすべてのプロセスのコアファイル名のパターンを設定します。

```
example$  coreadm -p core.%f.%p
```

プロセス ID が省略されているため、現在実行されているシェルでプロセスごとのコアファイル名パターンが設定され、それがすべての子プロセスに継承されます。

例 2 サブディレクトリへのユーザーのファイルのダンプ

次のコマンドでは、ユーザーのすべてのコアダンプを、システムノード名によって判別される、ホームディレクトリの corefiles サブディレクトリにダンプします。このコマンドは、多くの異なるマシンを使用し、共有ホームディレクトリを持つユーザーに役立ちます。

```
example$  coreadm -p $HOME/corefiles/%n.%f.%p 1234
```

例 3 大域コアファイルリポジトリの選別

次のコマンドでは、実行可能ファイルが /usr/bin または /usr/sbin から実行された場合にのみ大域リポジトリにコアファイルを生成するようにシステムを設定します。

```
example# mkdir -p /var/cores/usr/bin
example# mkdir -p /var/cores/usr/sbin
example# coreadm -G all -g /var/cores/%d/%f.%p.%n
```

ファイル

/var/cores
大域コアファイルの格納用に提供されているディレクトリです。

終了ステータス

次の終了値が返されます。

0
正常終了。

1
システムコアファイル構成の取得または変更中に致命的エラーが発生しました。

2
無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

gcore(1), pfexec(1), svcs(1), init(1M), svcadm(1M), exec(2), fork(2), setuid(2), time(2), syslog(3C), core(4), prof_attr(4), user_attr(4), attributes(5), smf(5)

注意事項

局所 (非大域) ゾーンでは、大域設定はそのゾーン内で実行中のプロセスに適用されます。また、大域ゾーンはすべてのゾーンで実行されるプロセスに適用されます。

大域設定とは、システムまたはゾーン全体に適用される設定のことで、必ずしも大域ゾーンで有効になる設定を意味するものではありません。

coreadm サービスはサービス管理機能 smf(5) により次のサービス識別子の下で管理されます。

```
svc:/system/coreadm:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

-g、-G、-i、-I、-e、および -d の各オプションは、solaris.smf.manage.coreadm および solaris.smf.value.coreadm の両方の承認が付与されているユーザー、役割、またはプロファイルも使用できます。

名前

cpustat – monitor system behavior using CPU performance counters

形式

```
cpustat -c eventspec [-c eventspec]... [-p period] [-T u | d ]
     [-sntD] [interval [count]]

cpustat -h
```

機能説明

The `cpustat` utility allows CPU performance counters to be used to monitor the overall behavior of the CPUs in the system.

If *interval* is specified, `cpustat` samples activity every *interval* seconds, repeating forever. If a *count* is specified, the statistics are repeated *count* times. If neither are specified, an interval of five seconds is used, and there is no limit to the number of samples that are taken.

オプション

The following options are supported:

`-c` *eventspec*

Specifies a set of events for the CPU performance counters to monitor. The syntax of these event specifications is:

```
[picn=]eventn[,attr[n][=val]][,[picn=]eventn
     [,attr[n][=val]],...,]
```

You can use the `-h` option to obtain a list of available events and attributes. This causes generation of the usage message. You can omit an explicit counter assignment, in which case `cpustat` attempts to choose a capable counter automatically.

Attribute values can be expressed in hexadecimal, octal, or decimal notation, in a format suitable for `strtoll(3C)`. An attribute present in the event specification without an explicit value receives a default value of `1`. An attribute without a corresponding counter number is applied to all counters in the specification.

The semantics of these event specifications can be determined by reading the CPU manufacturer's documentation for the events.

Multiple `-c` options can be specified, in which case the command cycles between the different event settings on each sample.

`-D`

Enables debug mode.

`-h`

Prints an extensive help message on how to use the utility and how to program the processor-dependent counters.

`-n`

Omits all header output (useful if `cpustat` is the beginning of a pipeline).

`-p` *period*

Causes `cpustat` to cycle through the list of *eventspec*s every *period* seconds. The tool sleeps after each cycle until *period* seconds have elapsed since the first *eventspec* was measured.

When this option is present, the optional *count* parameter specifies the number of total cycles to make (instead of the number of total samples to take). If *period* is less than the number of *eventspec*s times *interval*, the tool acts as it period is `0`.

`-s`
Creates an idle soaker thread to spin while system-only *eventspec*s are bound. One idle soaker thread is bound to each CPU in the current processor set. System-only *eventspec*s contain both the `nouser` and the `sys` tokens and measure events that occur while the CPU is operating in privileged mode. This option prevents the kernel's idle loop from running and triggering system-mode events.

`-T u | d`
Display a time stamp.

Specify `u` for a printed representation of the internal representation of time. See `time(2)`. Specify `d` for standard date format. See `date(1)`.

`-t`
Prints an additional column of processor cycle counts, if available on the current architecture.

使用法

A closely related utility, `cputrack(1)`, can be used to monitor the behavior of individual applications with little or no interference from other activities on the system.

The `cpustat` utility must be run by the super-user, as there is an intrinsic conflict between the use of the CPU performance counters system-wide by `cpustat` and the use of the CPU performance counters to monitor an individual process (for example, by `cputrack`.)

Once any instance of this utility has started, no further per-process or per-LWP use of the counters is allowed until the last instance of the utility terminates.

The times printed by the command correspond to the wallclock time when the hardware counters were actually sampled, instead of when the program told the kernel to sample them. The time is derived from the same timebase as `gethrtime(3C)`.

The processor cycle counts enabled by the `-t` option always apply to both user and system modes, regardless of the settings applied to the performance counter registers.

On some hardware platforms running in system mode using the “sys” token, the counters are implemented using 32-bit registers. While the kernel attempts to catch all overflows to synthesize 64-bit counters, because of hardware implementation restrictions, overflows can be lost unless the sampling interval is kept short enough. The events most prone to wrap are those that count processor clock cycles. If such an event is of interest, sampling should occur frequently so that less than 4 billion clock cycles can occur between samples.

The output of cpustat is designed to be readily parseable by `nawk(1)` and `perl(1)`, thereby allowing performance tools to be composed by embedding `cpustat` in scripts. Alternatively, tools can be constructed directly using the same APIs that `cpustat` is built upon using the facilities of `libcpc(3LIB)`. See `cpc(3CPC)`.

The cpustat utility only monitors the CPUs that are accessible to it in the current processor set. Thus, several instances of the utility can be running on the CPUs in different processor sets. See psrset(1M) for more information about processor sets.

Because cpustat uses LWPs bound to CPUs, the utility might have to be terminated before the configuration of the relevant processor can be changed.

使用例

SPARC

例 1 Measuring External Cache References and Misses

The following example measures misses and references in the external cache. These occur while the processor is operating in user mode on an UltraSPARC machine.

```
example% cpustat -c EC_ref,EC_misses 1 3

    time cpu event      pic0      pic1
   1.008   0  tick     69284      1647
   1.008   1  tick     43284      1175
   2.008   0  tick    179576      1834
   2.008   1  tick    202022     12046
   3.008   0  tick     93262       384
   3.008   1  tick     63649      1118
   3.008   2 total    651077     18204
```

x86

例 2 Measuring Branch Prediction Success on Pentium 4

The following example measures branch mispredictions and total branch instructions in user and system mode on a Pentium 4 machine.

```
 example% cpustat -c \
    pic12=branch_retired,emask12=0x4,pic14=branch_retired,\
    emask14=0xf,sys 1 3

    time cpu event      pic12     pic14
   1.010   1  tick       458       684
   1.010   0  tick       305       511
   2.010   0  tick       181       269
   2.010   1  tick       469       684
   3.010   0  tick       182       269
   3.010   1  tick       468       684
   3.010   2 total      2063      3101
```

例 3 Counting Memory Accesses on Opteron

The following example determines the number of memory accesses made through each memory controller on an Opteron, broken down by internal memory latency:

```
cpustat -c \
   pic0=NB_mem_ctrlr_page_access,umask0=0x01, \
   pic1=NB_mem_ctrlr_page_access,umask1=0x02, \
```

例 3 Counting Memory Accesses on Opteron (続き)

```
pic2=NB_mem_ctrlr_page_access,umask2=0x04,sys \
1

 time cpu event      pic0      pic1      pic2
1.003   0  tick     41976     53519      7720
1.003   1  tick      5589     19402       731
2.003   1  tick      6011     17005       658
2.003   0  tick     43944     45473      7338
3.003   1  tick      7105     20177       762
3.003   0  tick     47045     48025      7119
4.003   0  tick     43224     46296      6694
4.003   1  tick      5366     19114       652
```

警告

By running the `cpustat` command, the super-user forcibly invalidates all existing performance counter context. This can in turn cause all invocations of the `cputrack` command, and other users of performance counter context, to exit prematurely with unspecified errors.

If `cpustat` is invoked on a system that has CPU performance counters which are not supported by Solaris, the following message appears:

```
cpustat: cannot access performance counters - Operation not applicable
```

This error message implies that `cpc_open()` has failed and is documented in `cpc_open(3CPC)`. Review this documentation for more information about the problem and possible solutions.

If a short interval is requested, `cpustat` might not be able to keep up with the desired sample rate. In this case, some samples might be dropped.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	diagnostic/cpu-counters
Interface Stability	Committed

関連項目

`cputrack(1)`, `nawk(1)`, `perl(1)`, `iostat(1M)`, `prstat(1M)`, `psrset(1M)`, `vmstat(1M)`, `cpc(3CPC)`, `cpc_open(3CPC)`, `cpc_bind_cpu(3CPC)`, `gethrtime(3C)`, `strtoll(3C)`, `libcpc(3LIB)`, `attributes(5)`

注意事項

When `cpustat` is run on a Pentium 4 with HyperThreading enabled, a CPC set is bound to only one logical CPU of each physical CPU. See `cpc_bind_cpu(3CPC)`.

名前

croinfo, diskinfo – query and display information about 1) chassis, receptacle, and occupants or 2) disk occupants of bay receptacles

形式

```
croinfo [-h] [-v] [-I cro_db] [-o fields] [-O fields]
        [-P product-id] [-C chassis-id] [-A alias-id]
        [-R receptacle-name] [-T receptacle-type] [-t occupant-type]
        [-D devchassis-path] [-d occupant-devices] [-p occupant-paths]
        [-c occupant-compdev] [-i occupant-devid] [-m occupant-mfg]
        [-e occupant-model] [-n occupant-part] [-s occupant-serial]
        [-f occupant-firm] [-1 occupant-misc-1] [-2 occupant-misc-2]
        [-3 occupant-misc-3]

diskinfo [-h] [-v] [-I cro_db] [-o fields] [-O fields]
        [-P product-id] [-C chassis-id] [-A alias-id]
        [-R receptacle-name] [-T receptacle-type] [-t occupant-type]
        [-D devchassis-path] [-d occupant-devices] [-p occupant-paths]
        [-c occupant-compdev] [-i occupant-devid] [-m occupant-mfg]
        [-e occupant-model] [-n occupant-part] [-s occupant-serial]
        [-f occupant-firm] [-1 occupant-misc-1] [-2 occupant-misc-2]
        [-3 occupant-misc-3]

croinfo -?

diskinfo -?
```

機能説明

The `diskinfo` and `croinfo` utility share the same binary executable. At runtime, the utility checks to see how it was invoked, and adjusts defaults.

The `croinfo` utility allows users to query and display specific aspects of a system's configuration. Queries are performed against a record-oriented dataset that captures the relationship between physical location and various aspects of the device currently at that physical location. This relationship is expressed in terms of Chassis, Receptacle, and Occupant (thus the `cro` prefix).

Records in a CRO dataset are composed of multiple, named fields with each record having a potentially unique field value. An angle-bracket reference, such as *product-id*, is referring to a specific *field-name*. For a given record, a field value is either undefined (empty) or defined with a set of indexed string values. Some defined field values have just one string value, while others can have multiple string values.

Each *field-name* defined is associated with a separate *field-char* character. By convention, uppercase *field-char* characters are associated for chassis and receptacle information fields, and lowercase *field-char* characters are associated with occupant information fields. For each *field-char* character, a separate `-field-char field-name-RE` flag regular expression filter option is provided. This allows the user to customize queries to display information about specific aspects of the configuration. Records that match all regular expressions are selected, in dataset order, for display. For multiple string values, only one index value needs to match for the field to match.

By default, only a minimal number of default output fields are displayed. You can override the default with either `-o` *fields* for human-readable output or with `-O` *fields* for parseable output. In both cases, output fields can be specified using either the short-hand *field-char*[...] notation or in the more descriptive *field-name*[,...] notation.

For human-readable use, by using the `-o` *fields* option, the user can override the default fields, and output any fields, in the specified order, in a column-aligned whitespace separated format. In general, output will be one line of output per matching record with undefined (empty) *field-name* values displayed as a hyphen (–). If, however, a displayed record has a multiple string *value* field, then multiple lines of output are produced with any secondary non-multiple string values fields showing a colon (`:`).

For scripting, by using the `-O` *fields* option, the user can override the default output fields and output any fields, in the specified order, in a parseable colon-separated format with whitespace removed and column headers suppressed. Output will be one line of output per matching record with undefined (empty) field-name values displayed with no value. If a displayed record *field-name* has a multiple string *value* field, then all the values are concatenated, separated by a comma. Any occurrence of a colon or a comma in a value is escaped with a leading backslash (`\\`). To make scripts more legible, use of the `-O` *field-name*[,...]' notation is encouraged.

The `-o` and `-O` options are mutually exclusive.

If the `-h` option is used, or scripting output format is requested by using `-o`, the column headers for output fields are suppressed.

A Chassis is identified by a specific *product-id* and *chassis-id*. The *product-id* relates to a specific chassis-level product, like a system chassis Sun-Fire-X4200-M2 or a storage chassis SUN-Storage-J4410. For a given *product-id* value, the *chassis-id* defines a unique serial number.

A specific *product-id.chassis-id* combination can have an “managed” location-oriented *alias-id* defined by the administrator, using `fmadm(1M)` that provides installation-specific location information about where a chassis is physically located. This might include such information as building, room, rack, and U-number range within a rack.

In addition to the managed location-oriented *alias-id* defined using `fmadm(1M)`, system chassis always have one well-known alias called `SYS` that can be used to identify receptacles that are internal to the system chassis.

Within a chassis, each receptacle has a unique *receptacle-name* that should match the physical silk-screen label designation for the receptacle. Each receptacle also has a *receptacle-type*, which helps define acceptable *occupant-types*.

When a receptacle is occupied, use the `-f` flag definitions for available *occupant-information*. Of particular interest is the `-c` *occupant-compdev* occupant information: it describes the common component of the public `/dev` name associated with the occupant device. For disks, this is the whole-disk `c#t#d#` name.

The CRO dataset order is associated with the *devchassis-path* of the record, which corresponds to the `/dev/chassis` name space maintained by `devchassisd(1M)`. That ordering places records associated with the well-known `SYS` internal alias first, and records with `BOOT` receptacle-name first in `SYS`. This is done to ensure that, when applicable, information about the typical boot device is provided first.

For `croinfo`, the default output is in `-o` *Dtc* format, and all CRO records are shown.

For `diskinfo`, the default output is in `-o` *Dc* format, and a `-T` *bay receptacle-type* filter is applied. The meanings of the *occupant-misc-#* fields also take on a disk-specific interpretation: *misc-1* is capacity, and *misc-2* is target-port information. These defaults allow the `diskinfo` command to query the relationship between chassis, *bay* receptacles, and their *disk* occupants, while ignoring other CRO information.

オプション

For each record *field-name* defined, a separate *-field-char field-name-RE* flag regular expression filter can be specified. For a given *field-name*, if no specific *-field-char field-name-RE* filter is defined, all CRO records match.

This allows the user to customize queries to display information about specific aspects of the configuration. CRO records that match all of the specified *field-name* regular expressions (as in `regex(3C)`) will be selected for display, with specific fields output controlled by means of `-o`, `-O`, or the default.

`-P` *product-id*

The *product-id* specifies the product identifier of an enumerated chassis. The *product-id* might be exposed in the `/dev/chassis` name space. For storage products that do not have an established `fmadm(1M)` managed *alias-id*, the *product-id* is visible in the `devchassis(7FS)` `/dev/chassis` name space.

Example system *product-id* value: `Sun-Fire-X4200-M2`

Example storage *product-id* value: `SUN-Storage-J4410`

`-C` *chassis-id*

The *chassis-id* specifies the serial number of a product chassis. The *chassis-id* might be exposed in the `/dev/chassis` name space. For storage products that do not have an established `fmadm(1M)` managed *alias-id*, the *product-id* is visible in the `devchassis(7FS)` `/dev/chassis` name space.

Example *chassis-id* value: `0818QAJ002`

`-A` *alias-id*

An *alias-id* value can be the well-known alias value of `SYS`, for system internal devices. In addition, an *alias-id* value can be a managed alias, defined by the administrator using `fmadm(1M)`. The intended use of a managed alias is to define the physical location of the *product-id.chassis-id*. The *alias-id* is exposed in the `/dev/chassis` name space.

Example well-known *alias-id* value: `SYS`

Example managed *alias-id* value: `RACK29.U01-04`

`-R` *receptacle-name*
For a specific *product-id*, the unique *receptacle-name* defines location of a specific receptacle in a chassis. The *receptacle-name* should be identical to a silk-screen label on the physical chassis, and should also match product documentation. The *receptacle-name* can have multiple path components, such as `SYS/HD0`. The receptacle-name is exposed in the `/dev/chassis` name space.

Example *receptacle-name* value: `SYS/HD0`

`-T` *receptacle-type*
Example *receptacle-type* value: `bay`

`-t` *occupant-type*
A receptacle without an occupant has an undefined (empty) *occupant-type* value, shown as a hyphen (–).

Example *occupant-type* value: `disk`

`-D` *devchassis-path*
Example *devchassis-path* value: `/dev/chassis/SYS/HD0/disk`

`-d` *occupant-devices*
A receptacle without an occupant has an undefined (empty) *occupant-devices* value, shown as a hyphen (–).

Example *occupant-devices* value:

```
/devices/scsi_vhci/disk@g5000c500101ba0a3
```

`-p` *occupant-paths*
A receptacle without an occupant has an undefined (empty) *occupant-paths* value, shown as a hyphen (–).

Example occupant-paths value:

```
devices/pci@0,0/pci10de,5d@d/pci11f8,8001@0/iport@f/disk@w5000c500101ba0a1,0
```

`-c` *occupant-compdev*
A receptacle without an occupant has an undefined (empty) *occupant-compdev* value, shown as a hyphen (–).

Example *occupant-compdev* value: `c0t5000C500101BA0A3d0`

`-i` *occupant-devid*
A receptacle without an occupant has an undefined (empty) *occupant-devid* value, shown as a hyphen (–).

Example *occupant-devid* value: `id1,sd@n5000c500101ba0a3`

`-m` *occupant-mfg*
A receptacle without an occupant has an undefined (empty) *occupant-mfg* value, shown as a hyphen (–).

Example *occupant-mfg* value: `SEAGATE`

`-e` *occupant-model*
A receptacle without an occupant has an undefined (empty) *occupant-model*, shown as a hyphen (–).

Example *occupant-model* value: `ST32000SSSUN2.0T`

`-n` *occupant-part*
A receptacle without an occupant has an undefined (empty) *occupant-part* value, shown as a hyphen (–).

Example *occupant-part* value: `SEAGATE-ST32000SSSUN2.0T`

`-s` *occupant-serial*
A receptacle without an occupant has an undefined (empty) *occupant-serial* value, shown as a hyphen (–).

Example *occupant-serial* value: `000949L09C8L_____9WM09C8L`

`-f` *occupant-firm*
A receptacle without an occupant has an undefined (empty) *occupant-firm* value, shown as a hyphen (–).

Example *occupant-firm* value: `0313`

`-1` *occupant-misc-1*
A receptacle without an occupant has an undefined (empty) *occupant-misc-1* value, shown as a hyphen (–).

`-2` *occupant-misc-2*
A receptacle without an occupant has an undefined (empty) *occupant-misc-2* value, shown as a hyphen (–).

`-3` *occupant-misc-3*
A receptacle without an occupant has an undefined (empty) *occupant-misc-3* value, shown as a hyphen (–).

`-?`
Display usage information.

Output field-name Control Options

`-o` *fields*
Output specified fields, in order, in human-readable format.

For `croinfo`, default output is in `-o` *Dtc* format. For `diskinfo`, default output is in `-O` *Dc* format.

`-O` *fields*
Output specified fields, in order, in parseable format.

`-h`
Do not output *field-name* column headers

`-v`
Display verbose header that includes various information about when the CRO dataset was created. This option is of particular use with `-I` and is used to specify a non-standard source for the CRO dataset.

Dataset Selection Option

`-I` *cro_db*
Data file from which to obtain CRO dataset information.

使用例

In some of the following examples, example output wraps in an 80–character-wide display.

例 1 Determining Where a Disk is Located

The following command determines where a disk is located:

```
# croinfo -c c0t5000C500101BA0A3d0
D:devchassis-path                        t:occupant-type
---------------------------------------  ---------------
/dev/chassis/RACK29.U01-04/DISK_00/disk  disk

c:occupant-compdev
------------------
c0t5000C500101BA0A3d0
```

例 2 Reporting Internal Disks

The following command reports the *receptacle-name* and the *occupant-compdev* of internal disks, that is, disks that are associated with the well-known `SYS` alias.

```
# diskinfo -A SYS -o Rc
R:receptacle-name  c:occupant-compdev
-----------------  ------------------
SYS/HD0            c8t0d0
SYS/HD1            c8t1d0
SYS/HD2            -
SYS/HD3            -
```

Note that the `SYS/HD2` and `SYS/HD3` receptacles are empty.

The same command, in scripting output mode, would produce:

```
# diskinfo -A SYS -O receptacle-name,occupant-compdev
SYS/HD0:c8t0d0
SYS/HD1:c8t1d0
SYS/HD2:
SYS/HD3:
```

例 3 Reporting Disks in a Specific Enclosure

The following command reports all the disks within a chassis with a specific *product-id* and *chassis-id* value.

```
# diskinfo -P SUN-Storage-J4410 -C SUN-Storage-J4410 -o Rc
R:receptacle-name  c:occupant-compdev
-----------------  ---------------------
DISK_00            c0t5000C500101BA0A3d0
DISK_01            c0t5000C500101B95BBd0
DISK_02            -
```

例 4 Reporting Physical Path Information

The following command reports physical path information for a specific disk.

```
# croinfo -c c0t5000C500101BA0A3d0 -o cp
c:occupant-compdev
------------------
c0t5000C500101BA0A3d0

p:occupant-paths
--------------------------------------------------
/devices/pci@0,0/pci10de,5d@d/pci11f8,8001@0/iport@f/disk@w5000c500101ba0a1,0
/devices/pci@7b,0/pci10de,5d@d/pci11f8,8001@0/iport@f/disk@w5000c500101ba0a2,0
```

Note that *occupant-paths* has multiple string values.

例 5 Making Inventory of Disks

The following example reports how many of a specific type of disk are available using *occupant-part*.

```
# for i in ‘croinfo -h -o n  | sort -u‘
> do
> echo $i "\\t\\c";croinfo -h -n $i | wc -l
> done
SEAGATE-ST330055SSUN300G               3
SEAGATE-ST330056SSUN300G              19
SEAGATE-ST345056SSUN450G               5
```

例 6 Locating a Specific Type of Disk

The following command reports where disks of a specific type are located, what their `ctd` name is (by means of *occupant-compdev*), and what firmware level they are at.

```
# croinfo -n SEAGATE-ST330055SSUN300G -o Dcf
D:Devchassis                                       c:component
-----------------------------------------------  ---------------------
/dev/chassis/RACK29.U29-32/SCSI_Device__11/disk  c0t5000C50007DD49F7d0
/dev/chassis/RACK29.U33-36/SCSI_Device__18/disk  c0t5000C50008F7FB4Fd0
/dev/chassis/RACK29.U33-36/SCSI_Device__19/disk  c0t5000C50007DD412Fd0
```

例 6 Locating a Specific Type of Disk (続き)

```
f:firm
------
0892
0892
0892
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below

The interface stability of croinfo and diskinfo is Committed. The interface stability of command output is Not-an-Interface.

関連項目

devchassisd(1M), fmadm(1M), fmd(1M), attributes(5), devchassis(7FS)

The SCSI Storage Interfaces committee website, http://www.t10.org

SCSI Primary Commands-4, SPC4; SCSI Enclosure Services-2; SES2, Serial Attached SCSI-2, SAS2

注意事項

croinfo representation depends on the ability of fmd(1M) to enumerate system topology and accurately represent associated chassis, receptacles, and occupants. These dependencies might extend through fmd(1M) and require that connected hardware, and its associated firmware, comply with specific standards. For disk bays, this requires that storage chassis behave in a T10 standards-compliant (SPC4 and SES2) fashion. Storage chassis that do not respond appropriately might not report chassis, bays, or disk nodes correctly. Specifically, diskinfo requires that chassis support SES diagnostic page 0xa (Additional Element Status) and set the Element Index Present (EIP) bit to 1. Enclosures that do not meet this criterion will not be fully enumerated, and thus will not be properly represented.

名前

cron – クロックデーモン

形式

/usr/sbin/cron

機能説明

cron は、指定された日時にコマンドを実行するプロセスを開始します。

/var/spool/cron/crontabs ディレクトリの crontab ファイルにある指示に従って、定期的にスケジュールされたコマンドを cron に指定できます。crontab(1) コマンドを使用すると、ユーザー固有の crontab ファイルを生成できます。一度だけ実行するコマンドは、at(1) コマンドを使用して投入できます。

cron は、それ自体のプロセス初期化フェーズの際、および crontab または at コマンドが実行されるときにのみ、crontab または at コマンドファイルを検査します。これにより、定期的にスケジュールされた間隔で新しいファイルまたは変更されたファイルをチェックするオーバーヘッドが減少します。

cron は終了することがないため、一度だけ実行してください。これは、通常の手順では svc:/system/cron:default サービスによって実行されているものです。複数の cron インスタンスが実行されないように、ロックファイルとして /etc/cron.d/FIFO ファイルが使用されます。

cron は、ジョブの stdout および stderr ストリームの出力を取得し、空でない場合は、出力をユーザーにメールで送信します。ジョブが出力を生成しなかった場合、ユーザーにメールは送信されません。例外は、ジョブが at(1) ジョブで、ジョブが実行されたときに -m オプションが指定された場合です。

ユーザーのアカウントがロックされていると、cron ジョブおよび at ジョブは実行されません。ジョブとプロセスは実行されます。shadow(4) ファイルにはロックされていないアカウントが定義されており、それらのアカウントのジョブとプロセスは実行されます。

複数のタイムゾーンの cron ジョブを設定

cron デーモンのタイムゾーンは、cron エントリのシステム全体のタイムゾーンとして設定されます。また、さらに /etc/default/init を使用してシステム全体のデフォルトタイムゾーンとして設定されます。cron エントリのタイムゾーンはユーザーの crontab ファイルで上書きできます。crontab(1) を参照してください。

夏時間や冬時間などが有効になっている場合には、切り替え期間に予定されているジョブは、1 回だけ実行されたり、2 回実行されたり、またはまったく実行されないこともあります。

cron のデフォルトの設定

cron によって実行されるすべての操作のログを保存するには、/etc/default/cron ファイルで CRONLOG=YES を指定する必要があります。CRONLOG=NO を指定すると、ログの記録は行われません。通常、cron は非常に大きなログファイルを作成するため、ログの保存はユーザーが構成可能なオプションとなっています。

ユーザーの cron ジョブの PATH を設定するには、/etc/default/cron に PATH= を指定します。root ユーザーの cron ジョブの PATH を設定するには、/etc/default/cron に SUPATH = を指定します。PATH および SUPATH を設定した場合のセキュリティーへの影響を十分に考慮してください。

/etc/default/cron ファイルの例

```
CRONLOG=YES
PATH=/usr/bin:/usr/ucb:
```

この例のファイルは、ログの記録を有効にし、root ユーザー以外のジョブによって使用されるデフォルトの PATH を /usr/bin:/usr/ucb: に設定します。root ユーザーのジョブは、引き続き /usr/sbin:/usr/bin を使用します。

cron のログファイルは、logadm(1M) によって定期的にローテーションされます。

ファイル

/etc/cron.d	主 cron ディレクトリ
/etc/cron.d/FIFO	ロックファイル
/etc/default/cron	cron のデフォルト設定ファイル
/var/cron/log	cron の履歴情報
/var/spool/cron	スプール領域
/etc/cron.d/queuedefs	at、batch、および cron の待ち行列記述ファイル
/etc/logadm.conf	logadm の構成ファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

svcs(1), at(1), crontab(1), sh(1), logadm(1M), svcadm(1M), queuedefs(4), shadow(4), attributes(5), rbac(5), smf(5), smf_security(5)

注意事項

cron サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/cron:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。ほとんどの管理操作は、solaris.smf.manage.cron 承認を持つユーザーに委任できます (rbac(5) および smf_security(5) を参照)。

診断

cron によって行われたすべての操作の履歴は、/var/cron/log および場合によっては /var/cron/olog に格納されます。

名前

cryptoadm – 暗号化フレームワークの管理

形式

```
cryptoadm list [-mpv] [provider=provider-name]
     [mechanism=mechanism-list]

cryptoadm disable
     provider=provider-name mechanism=mechanism-list | random | all

cryptoadm enable
     provider=provider-name mechanism=mechanism-list | random | all

cryptoadm install provider=provider-name

cryptoadm install provider=provider-name
     [mechanism=mechanism-list]

cryptoadm uninstall provider=provider-name

cryptoadm unload provider=provider-name

cryptoadm disable fips-140

cryptoadm enable fips-140

cryptoadm list fips-140

cryptoadm refresh

cryptoadm start

cryptoadm stop

cryptoadm --help
```

機能説明

`cryptoadm` ユーティリティーは、システムの暗号化プロバイダ情報の表示、各プロバイダのメカニズムポリシーの構成、および暗号化プロバイダのインストールとアンインストールを行います。暗号化フレームワークでは、ユーザーレベルプロバイダ (PKCS11 共有ライブラリ)、カーネルソフトウェアプロバイダ (読み込み可能なカーネルソフトウェアモジュール)、およびカーネルハードウェアプロバイダ (暗号化ハードウェアデバイス) の 3 種類のプロバイダがサポートされます。

カーネルソフトウェアプロバイダ用に、`cryptoadm` ユーティリティーには `unload` サブコマンドが用意されています。このサブコマンドは、カーネルソフトウェアプロバイダを読み込み解除するようにカーネルに指示します。

暗号化フレームワークのメタスロット用に、`cryptoadm` ユーティリティーには、メタスロットの機能を有効/無効にする、メタスロットの構成を一覧表示する、代替の永続オブジェクトストレージを指定する、およびメタスロットのメカニズムポリシーを構成するための各サブコマンドが用意されています。

`cryptoadm` ユーティリティーには、暗号化フレームワークで FIPS-140 モードを有効/無効にするためのサブコマンドが用意されています。また、FIPS-140 モードの現在のステータスを表示するための `list` サブコマンドも用意されています。

管理者は、暗号化サブシステムの管理に syslog 機能 (syslogd(1M) および logadm(1M) を参照) を使用すると便利です。ログは、次のような状況で特に役立ちます。

- カーネルレベルのデーモンが停止し、すべてのアプリケーションでエラーが発生した場合。この状況を syslog から把握し、svcadm(1M) を使用して svc:/system/cryptosvc サービスをリブートできます。
- 問題のあるプロバイダがフレームワークに接続されている場合、この状況を syslog から把握して、問題のあるプロバイダをフレームワークから削除できます。

下記のサブコマンドまたはオプションを除き、cryptoadm コマンドは特権ユーザーが実行する必要があります。

- サブコマンド list、任意のオプション
- サブコマンド --help

オプション

cryptoadm ユーティリティーには、次に示すようなサブコマンドとオプションのさまざまな組み合わせがあります。

cryptoadm list
: インストールされているプロバイダの一覧を表示します。

cryptoadm list metaslot
: メタスロットのシステム全体の構成を表示します。

cryptoadm list -m [provider=*provider-name* | metaslot]
: インストールされているプロバイダまたはメタスロットで使用できるメカニズムの一覧を表示します。プロバイダを指定した場合、指定されたプロバイダの名前と、そのプロバイダで使用できるメカニズムの一覧を表示します。metaslot キーワードを指定した場合、メタスロットで使用できるメカニズムの一覧を表示します。

cryptoadm list -p [provider=*provider-name* | metaslot]
: インストールされているプロバイダのメカニズムポリシー (つまり、どのメカニズムが使用できて、どのメカニズムが使用できないか) を表示します。プロバイダ機能のポリシーまたはメタスロットも表示します。プロバイダを指定した場合、プロバイダの名前と、そのプロバイダにのみ適用されているメカニズムポリシーを表示します。metaslot キーワードを指定した場合、メタスロットに適用されているメカニズムポリシーを表示します。

cryptoadm list -v provider=*provider-name* | metaslot
: プロバイダを指定した場合、指定されたプロバイダに関する詳細情報を表示します。metaslot キーワードを指定した場合、メタスロットに関する詳細情報を表示します。

-v
: 上記のさまざまな list サブコマンド (list -p を除く) で、-v (詳細) オプションは、プロバイダ、メカニズム、およびスロットの詳細情報を表示します。

`cryptoadm disable provider=`*provider-name*
[mechanism=*mechanism-list* | *provider-feature* `...` | `all`]
プロバイダに指定されているメカニズムまたはプロバイダの機能を無効にします。*mechanism*、*provider-feature*、および `all` キーワードの説明については、「オペランド」の節を参照してください。

`cryptoadm [ mechanism=`*mechanism-list* `] [ auto-key-migrate ]`
暗号化フレームワークでメタスロット機能を無効にするか、またはメタスロット機能の一部を無効にします。オペランドを指定しない場合、このコマンドは、暗号化フレームワークでメタスロット機能を無効にします。メカニズムのリストを指定した場合、メタスロットの指定されたメカニズムを無効にします。メタスロットのすべてのメカニズムを無効にした場合、メタスロットは無効化されます。メカニズムの説明については、「オペランド」の節を参照してください。`auto-key-migrate` キーワードを指定した場合、暗号化操作を実行するために必要な場合であっても、機密性の高いトークンオブジェクトのほかのスロットへの移行が無効になります。`auto-key-migrate` の説明については、「オペランド」の節を参照してください。

`cryptoadm enable provider=`*provider-name*
[mechanism=*mechanism-list* | *provider-feature* `...` | `all`]
プロバイダに指定されているメカニズムまたはプロバイダの機能を有効にします。*mechanism*、*provider-feature*、および `all` キーワードの説明については、「オペランド」の節を参照してください。

`cryptoadm enable metaslot [ mechanism=`*mechanism-list* `] |`
`[ [ token=`*token-label*`] [ slot=`*slot-description*`] |`
`default-keystore ] | [ auto-key-migrate ]`
オペランドを指定しない場合、このコマンドは、暗号化フレームワークでメタスロット機能を有効にします。メカニズムのリストを指定した場合、メタスロットの一連の指定されたメカニズムのみを有効にします。*token-label* を指定した場合、指定したトークンが永続オブジェクトストアとして使用されます。*slot-description* を指定した場合、指定したスロットが永続オブジェクトストアとして使用されます。*token-label* と *slot-description* の両方を指定した場合、一致するトークンラベルとスロットの説明を持つプロバイダが永続オブジェクトストアとして使用されます。`default-keystore` キーワードを指定した場合、メタスロットでデフォルトの永続オブジェクトストアが使用されます。`auto-key-migrate` キーワードを指定した場合、特定の暗号化操作を完了するために、必要に応じて機密性の高いトークンオブジェクトがほかのスロットに自動的に移行されます。メカニズム、トークン、スロット、`default-keystore`、および `auto-key-migrate` の説明については、「オペランド」の節を参照してください。

`cryptoadm install provider=`*provider-name*
ユーザーレベルプロバイダをシステムにインストールします。*provider* オペランドは、対応する共有ライブラリの絶対パス名である必要があります。ライブラリの 32 ビットバージョンと 64 ビットバージョンの両方がある場合は、`$ISA` を含む

パス名を使用してこのコマンドを 1 回だけ実行するようにしてください。`$ISA` は、環境変数への参照ではありません。また、`$ISA` を引用符で囲むか (単一引用符を使用、[例: `'$ISA'`])、`$` をエスケープして、シェルによって誤って展開されないようにする必要があります。ユーザーレベルフレームワークでは、`$ISA` が空の文字列またはアーキテクチャー固有のディレクトリ (例: `sparcv9`) に展開されます。

ユーザーレベルプロバイダをインストールする場合は、プロバイダのパッケージをビルドすることをお勧めします。詳細は、『Solaris セキュリティーサービス開発ガイド』を参照してください。

`cryptoadm install provider=`*provider-name*
mechanism=*mechanism-list*
カーネルソフトウェアプロバイダをシステムにインストールします。provider には、ベース名のみを含めるようにしてください。*mechanism-list* オペランドには、このプロバイダでサポートするメカニズムの完全なリストを指定します。

カーネルソフトウェアプロバイダをインストールする場合は、プロバイダのパッケージをビルドすることをお勧めします。詳細は、『Solaris セキュリティーサービス開発ガイド』を参照してください。

`cryptoadm uninstall provider=`*provider-name*
指定された *provider* と、関連付けられているメカニズムポリシーをシステムからアンインストールします。このサブコマンドは、ユーザーレベルプロバイダまたはカーネルソフトウェアプロバイダのみに適用されます。

`cryptoadm unload provider=`*provider-name*
provider によって指定されたカーネルソフトウェアモジュールを読み込み解除します。

`cryptoadm disable fips-140`
暗号化フレームワークおよびハードウェアプロバイダにおいて FIPS-140 モードを無効にします。

`cryptoadm enable fips-140`
暗号化フレームワークおよびハードウェアプロバイダにおいて FIPS-140 モードを有効にします。このサブコマンドは、ユーザーレベル `pkcs11_softtoken` ライブラリおよびカーネルソフトウェアプロバイダの FIPS 未承認アルゴリズムは無効にしません。FIPS 承認アルゴリズムのみを使用するかどうかは、フレームワークの消費者しだいです。

このサブコマンドが完了すると、境界外に追加されたプラグインによって FIPS への準拠が無効化される可能性があること、およびそれらのプラグインのセキュリティーポリシーの確認が必要というメッセージが管理者に表示されます。

暗号化アルゴリズムテストおよびソフトウェア完全性テストを含む電源投入時自己診断を実行するために、システムをリブートする必要があります。

`cryptoadm list fips-140`
暗号化フレームワークおよびハードウェアプロバイダにおける FIPS-140 モードの現在の設定を表示します。FIPS-140 モードのステータスは、`enabled` または `disabled` です。デフォルトの FIPS-140 モードは、`disabled` です。

`cryptoadm refresh`
`cryptoadm start`
`cryptoadm stop`
`smf(5)` によって使用される非公開インタフェースです。これらは直接使用しないでください。

`cryptoadm -help`
コマンドの使用法を表示します。

オペランド

provider=*provider-name*
ユーザーレベルプロバイダ (PKCS11 共有ライブラリ)、カーネルソフトウェアプロバイダ (読み込み可能なカーネルソフトウェアモジュール)、またはカーネルハードウェアプロバイダ (暗号化ハードウェアデバイス) です。

provider オペランドの有効な値は、`cryptoadm` *list* 形式のコマンドの出力からの 1 つのエントリです。ユーザーレベルプロバイダの場合の *provider* オペランドは、対応する共有ライブラリの絶対パス名です。カーネルソフトウェアプロバイダの場合の *provider* オペランドには、ベース名のみを含めます。カーネルハードウェアプロバイダの場合の *provider* オペランドは、「*name*/*number*」の形式で指定します。

mechanism=*mechanism-list*
1 つ以上の PKCS #11 メカニズムのコンマ区切りのリストです。PKCS #11 の仕様で定義されている暗号化操作を実装するプロセスです。*mechanism-list* の代わりに `all` を使用することで、プロバイダのすべてのメカニズムを指定できます。下記の `all` キーワードについての説明を参照してください。

provider-feature
所定のプロバイダの暗号化フレームワークの機能です。現時点では、`random` のみが機能として受け入れられます。ユーザーレベルプロバイダの場合、ランダム機能を無効にすると、PKCS #11 ルーチン `C_GenerateRandom` および `C_SeedRandom` がプロバイダから使用できなくなります。カーネルプロバイダの場合、ランダム機能を無効にすると、`/dev/random` でプロバイダから乱数を収集できなくなります。

`all`
キーワード `all` を `disable` サブコマンドおよび `enable` サブコマンドとともに使用すると、すべてのプロバイダ機能に対して実行できます。

`token=`*token-label*
暗号化フレームワークにおけるいずれかのプロバイダのトークンのラベルです。

token オペランドの有効な値は、コマンド `cryptoadm list -v` の出力の「Token Label」に示されている項目です。

`slot=`*slot-description*
暗号化フレームワークにおけるいずれかのプロバイダのスロットの説明です。

slot オペランドの有効な値は、コマンド `cryptoadm list -v` の出力の「機能説明」に示されている項目です。

`default-keystore`
キーワード `default-keystore` は、メタスロットの場合のみ有効です。このキーワードを指定すると、メタスロットの永続オブジェクトストアの設定が、デフォルトのストアを使用するように戻されます。

`auto-key-migrate`
キーワード auto-key-migrate は、メタスロットの場合のみ有効です。このキーワードを指定して、暗号化操作を実行するために、メタスロットで、機密性の高いトークンオブジェクトをトークンオブジェクトスロットからほかのスロットに移動できるようにするかどうかを構成します。

キーワード `all` は、2 とおりの方法で `disable` サブコマンドおよび `enable` サブコマンドと組み合わせて使用できます。

- 次のように、`all` を `mechanism=`*mechanism-list* およびその他の provider-feature の代わりに使用できます。

  ```
  # cryptoadm enable provider=dca/0 all
  ```

  次のコマンドでは、プロバイダのメカニズムおよびその他の provider-feature (`random` など) が有効になります。

- 次のように、`all` を `mechanism` の引数として使用することもできます。

  ```
  # cryptoadm enable provider=des mechanism=all
  ```

  このコマンドでは、プロバイダのすべてのメカニズムが有効になりますが、ほかの provider-feature (`random` など) は有効にはなりません。

使用例

例 1　システムにインストールされているプロバイダの一覧の表示

次のコマンドは、インストールされているすべてのプロバイダの一覧を表示します。

```
example% cryptoadm list
user-level providers:
/usr/lib/security/$ISA/pkcs11_kernel.so
/usr/lib/security/$ISA/pkcs11_softtoken.so
/opt/lib/libcryptoki.so.1
/opt/system/core-osonn/lib/$ISA/libpkcs11.so.1

kernel software providers:
    des
    aes
    bfish
```

例 1 システムにインストールされているプロバイダの一覧の表示 *(続き)*

```
    sha1
    md5

kernel hardware providers:
    dca/0
```

例 2 md5 プロバイダのメカニズムの一覧の表示

次のコマンドは、list サブコマンドの一種です。

```
example% cryptoadm list -m provider=md5
md5: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL
```

例 3 カーネルソフトウェアプロバイダの特定のメカニズムの無効化

次のコマンドは、カーネルソフトウェアプロバイダ des のメカニズム CKM_DES3_ECB および CKM_DES3_CBC を無効にします。

```
example# cryptoadm disable provider=des
```

例 4 プロバイダのメカニズムポリシーの表示

次のコマンドは、des プロバイダのメカニズムポリシーを表示します。

```
example% cryptoadm list -p provider=des
des: All mechanisms are enabled, except CKM_DES3_ECB, CKM_DES3_CBC
```

例 5 プロバイダの特定のメカニズムの有効化

次のコマンドは、カーネルソフトウェアプロバイダ des の CKM_DES3_ECB メカニズムを有効にします。

```
example# cryptoadm enable provider=des mechanism=CKM_DES3_ECB
```

例 6 ユーザーレベルプロバイダのインストール

次のコマンドは、ユーザーレベルプロバイダをインストールします。

```
example# cryptoadm install provider=/opt/lib/libcryptoki.so.1
```

例 7 32 ビットバージョンと 64 ビットバージョンを含むユーザーレベルプロバイダのインストール

次のコマンドは、32 ビットバージョンと 64 ビットバージョンの両方を含むユーザーレベルプロバイダをインストールします。

```
example# cryptoadm install \
provider=/opt/system/core-osonn/lib/'$ISA'/libpkcs11.so.1
```

例 8 プロバイダのアンインストール

次のコマンドは、md5 プロバイダをアンインストールします。

```
example# cryptoadm uninstall provider=md5
```

例 9 メタスロットの無効化

次のコマンドは、暗号化フレームワークでメタスロット機能を無効にします。

```
example# cryptoadm disable metaslot
```

例 10 指定されたトークンを永続オブジェクトストアとして使用するようにメタスロットを指定する

次のコマンドは、メタスロットが Venus トークンを永続オブジェクトストアとして使用するように指定します。

```
example# cryptoadm enable metaslot token="SUNW,venus"
```

終了ステータス

次の終了値が返されます。

0
: 正常終了。

>0
: エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照。

start、stop、および refresh の各オプションは、非公開インタフェースです。その他すべてのオプションおよびユーティリティー名は「確実」です。

関連項目

logadm(1M), svcadm(1M), syslogd(1M), libpkcs11(3LIB), exec_attr(4), prof_attr(4), attributes(5), smf(5), random(7D)

『Oracle Solaris の管理: セキュリティーサービス』

『Solaris セキュリティーサービス開発ガイド』

注意事項

ハードウェアプロバイダのポリシーが明示的になっていて (つまり、そのメカニズムの一部が無効になっていて)、ハードウェアプロバイダが切り離されていた場合、このハードウェアプロバイダのポリシーは、引き続き一覧に表示されます。

`cryptoadm` では、少なくとも、32 ビット共有オブジェクトが各ユーザーレベルプロバイダに提供されていることを前提としています。32 ビットおよび 64 ビット共有オブジェクトの両方が提供されている場合、2 つのバージョンが同じ機能を提供している必要があります。両方に同じメカニズムポリシーが適用されます。

名前

cvcd – 仮想コンソールデーモン

形式

```
/platform/platform_name/cvcd [-a auth] [-e encr]
    [-u esp_auth]
```

機能説明

仮想コンソールデーモン cvcd は、一部のプラットフォームで提供されているネットワークコンソールをサポートするサーバープロセスです。cvcd デーモンは、遠隔ホストからのネットワークコンソールの接続を受け入れます (一回につき 1 つのホストのみ)。コンソールの入力はこの接続から読み取られ、cvcredir(7D) を経由して cvc(7D) へ転送されます。

同様に、コンソールの出力は cvcredir(7D) から読み取られ、ネットワークコンソールの接続を介して転送されます。cvcd が終了した場合は、コンソールトラフィックは自動的に内部ハードウェアインタフェースによってルート変更されます。

通常、cvcd デーモンはシステムブート時にブートされます。各ドメインで同時に実行することができる cvcd プロセスは、1 つだけです。

注意 – Sun Enterprise 10000 ドメインでは、cvcd は構成ファイル (/etc/ssphostname) を使用して、ネットワークコンソールの接続が許可されるホストの名前を決定します。遠隔コンソールホストの名前が変更された場合は、構成ファイルを編集し、その変更点を反映させる必要があります。

cvcd デーモンは、次に説明する各オプションを通じてソケット単位の IP セキュリティーアーキテクチャー (IPsec) をサポートします。ipsec(7P) を参照してください。

オプション

cvcd デーモンがサポートするオプションは、次のとおりです。

-a *auth*	IPsec 認証ヘッダー (AH) アルゴリズムを制御します。*auth* に指定できるのは、none、md5、sha1 のいずれかです。
-e *encr*	IPsec ESP (Encapsulating Security Payload) 暗号化アルゴリズムを制御します。*encr* に指定できるのは、none、des、3des のいずれかです。
-u *esp_auth*	IPsec ESP (Encapsulating Security Payload) 認証アルゴリズムを制御します。*esp_auth* に指定できるのは、none、md5、sha1 のいずれかです。

オペランド

次のオペランドがサポートされています。

platform_name	パッケージおよびコードで使用されている、Sun の正式なプラットフォーム名。たとえば、Sun Fire 15000 サーバーの場合は *platform_name* は SUNW,Sun-Fire-15000 になります。

使用例

例 1　IPSec オプションの設定

次のコマンドは、IPsec 認証ヘッダーアルゴリズムの値を md5 に設定します。このコマンドの結果として、cvcd は HMAC-MD5 認証アルゴリズムを使用するようになります。

例 1　IPSec オプションの設定　　(続き)

```
# svccfg -s svc:/system/cvc setprop cvc/ah_auth = "md5"
# svccfg -s svc:/system/cvc setprop cvc/esp_encr = "none"
# svccfg -s svc:/system/cvc setprop cvc/esp_auth = "none"
# svcadm refresh svc:/system/cvc
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	Sun Enterprise 10000 サーバー、Sun Fire High-End Systems
使用条件	system/network-console

関連項目

svcs(1), svcadm(1M), svccfg(1M), services(4), attributes(5), smf(5), cvc(7D), cvcredir(7D), ipsec(7P)

『Sun Enterprise 10000 SSP リファレンスマニュアル』

『System Management Services (SMS) リファレンスマニュアル』

注意事項

cvcd サービスは、サービス管理機能 smf(5) によって、次の障害管理リソース識別子 (FMRI) の下で管理されます。

```
svc:/system/cvc
```

有効化、無効化、再起動要求など、このサービスに関する管理操作は、svcadm(1M) または svccfg(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

datadm – maintain DAT static registry file

形式

```
/usr/bin/datadm [-v] [-u] [-a service_provider.conf]
     [-r service_provider.conf]
```

機能説明

The `datadm` utility maintains the DAT static registry file, `dat.conf(4)`.

This administrative configuration program allows uDAPL service providers to add and remove themselves to the `dat.conf` file.

You can add or remove interface adapters that a service provider supports from a system after its installation. You can use `datadm` to update the `dat.conf` file to reflect the current state of the system. A new set of interface adapters for all the service providers currently installed is regenerated.

オプション

The following options are supported:

`-a service_provider.conf`
: Enumerate each device entry in the `service_provider.conf(4)` file into a list of interface adapters, that is, interfaces to external network that are available to uDAPL consumers.

`-r service_provider.conf`
: Remove the list of interface adapters that corresponds to the device entry in the `service_provider.conf(4)` file.

`-u`
: Update the `dat.conf` to reflect the current state of the system with an up to date set of interface adapters for the service providers that are currently listed in the DAT static registry.

`-v`
: Display the DAT static registry file, `dat.conf`.

使用例

例 1　Enumerating a Device Entry

The following example enumerates a device entry in the `service_provider.conf(4)` file into interface adapters in the `dat.conf(4)` file.

Assume that `SUNW` has a service provider library that supports the device `tavor`. It has a `service_provider.conf(4)` file installed in the directory `/usr/share/dat/SUNWudaplt.conf` with a single entry as follows:

```
driver_name=tavor u1.2 nonthreadsafe default\
    udapl_tavor.so.1 SUNW.1.0 ""
```

`tavor` is an Infiniband Host Channel Adapter with two ports. Both IB ports exist in a single IB partition, `0x8001`. If an IB partition is created and plumbed to each port (with the names `p8001.ibd0` and `p8001.ibd1`), there will be two IB partition instances. See `dladm(1M)` for more information on creating IB partition data links.

例 1 Enumerating a Device Entry (続き)

```
# dladm show-part

LINK            PKEY      OVER       STATE      FLAGS
p8001.ibd0      8001      ibd0       unknown    ----
p8001.ibd1      8001      ibd1       unknown    ----
```

Running the command:

```
# datadm -a /usr/share/dat/SUNWudaplt.conf
```

...appends two new entries (if they do not already exist) in the `/etc/dat/dat.conf` file:

```
p8001.ibd0 u1.2 nonthreadsafe default udapl_tavor.so.1 SUNW.1.0 ""
"driver_name=tavor"
p8001.ibd1 u1.2 nonthreadsafe default udapl_tavor.so.1 SUNW.1.0 ""
"driver_name=tavor"
```

例 2 Updating the dat.conf to Reflect the Current State of the System

A new IB partition, `0x8002`, is added to the above example covering port 1 of the Host Channel Adapter. If a new IB partition is created on the port 1/partition `0x8002` with the partition link name specified as `p8002.ibd0`, there will be a third IB partition instance: `p8002.ibd0`.

```
# dladm show-part
LINK            PKEY      OVER       STATE      FLAGS
p8001.ibd0      8001      ibd0       unknown    ----
p8001.ibd1      8001      ibd1       unknown    ----
p8002.ibd0      8000      ibd0       unknown    ----
```

Running `datadm -u` command, updates the `/etc/dat/dat.conf` file with a new entry added reflecting the current state of the system.

`datadm -v` shows that there are now three entries in the `/etc/dat/dat.conf` file:

```
p8001.ibd0 u1.2 nonthreadsafe default udapl_tavor.so.1 SUNW.1.0 ""
"driver_name=tavor"
p8001.ibd1 u1.2 nonthreadsafe default udapl_tavor.so.1 SUNW.1.0 ""
"driver_name=tavor"
p8002.ibd0 u1.2 nonthreadsafe default udapl_tavor.so.1 SUNW.1.0 ""
"driver_name=tavor"
```

ファイル

`/etc/dat/dat.conf`
DAT static registry file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/io/infiniband/udapl
Interface Stability	Committed

関連項目

dladm(1M), pkgadd(1M), pkgrm(1M), libdat(3LIB), dat.conf(4), service_provider.conf(4), attributes(5)

名前

dcs – domain configuration server

形式

```
/usr/lib/dcs [-s sessions]
     [ [-a auth] [-e encr] [-u esp_auth] ] [-l]
```

機能説明

The Domain Configuration Server (DCS) is a daemon process that runs on Sun servers that support remote Dynamic Reconfiguration (DR) clients. It is started by the Service Management Facility (see smf(5)) when the first DR request is received from a client connecting to the network service sun-dr. After the DCS accepts a DR request, it uses the libcfgadm(3LIB) interface to execute the DR operation. After the operation is performed, the results are returned to the client.

The DCS listens on the network service labeled sun-dr. Its underlying protocol is TCP. It is invoked as a server program by the SMF using the TCP transport. The fault management resource identifier (FMRI) for DCS is:

```
svc:/platform/sun4u/dcs:default
```

If you disable this service, DR operations initiated from a remote host fail. There is no negative impact on the server.

Security for the DCS connection is provided differently based upon the architecture of the system. The SMF specifies the correct options when invoking the DCS daemon, based upon the current architecture. For all architectures, security is provided on a per-connection basis.

The DCS daemon has no security options that are applicable when used on a Sun Enterprise 10000 system. So there are no options applicable to that architecture.

The security options for Sun Fire high-end systems are based on IPsec options defined as SMF properties. These options include the -a *auth*, -e *encr*, and -u *esp_auth* options, and can be set using the svccfg(1M) command. These options must match the IPsec policies defined for DCS on the system controller. Refer to the kmd(1M) man page in the System Management Services (SMS) Reference Manual. The kmd(1M) man page is not part of the SunOS man page collection.

Security on SPARC Enterprise Servers is not configurable. The DCS daemon uses a platform-specific library to configure its security options when running on such systems. The -l option is provided by the SMF when invoking the DCS daemon on SPARC Enterprise Servers. No other security options to the DCS daemon should be used on SPARC Enterprise Servers.

オプション

The following options are supported:

-a *auth* — Controls the IPsec Authentication Header (AH) algorithm. *auth* can be one of none, md5, or sha1.

-e *encr* — Controls the IPsec Encapsulating Security Payload (ESP) encryption algorithm. *encr* can be one of none, des, or 3des.

`-l`	Enables the use of platform-specific security options on SPARC Enterprise Servers.
`-s` *sessions*	Sets the number of active sessions that the DCS allows at any one time. When the limit is reached, the DCS stops accepting connections until active sessions complete the execution of their DR operation. If this option is not specified, a default value of 128 is used.
`-u` *esp_auth*	Controls the IPsec Encapsulating Security Payload (ESP) authentication algorithm. *esp_auth* can be one of `none`, `md5`, or `sha1`.

使用例

例 1 Setting an IPSec Option

The following command sets the Authentication Header algorithm for the DCS daemon to use the HMAC-MD5 authentication algorithm. These settings are only applicable for using the DCS daemon on a Sun Fire high-end system.

```
# svccfg -s svc:/platform/sun4u/dcs setprop dcs/ah_auth = "md5"
# svccfg -s svc:/platform/sun4u/dcs setprop dcs/esp_encr = "none"
# svccfg -s svc:/platform/sun4u/dcs setprop dcs/esp_auth = "none"
# svcadm refresh svc:/platform/sun4u/dcs
```

エラー

The DCS uses `syslog(3C)` to report status and error messages. All of the messages are logged with the `LOG_DAEMON` facility. Error messages are logged with the `LOG_ERR` and `LOG_NOTICE` priorities, and informational messages are logged with the `LOG_INFO` priority. The default entries in the `/etc/syslog.conf` file log all of the DCS error messages to the `/var/adm/messages` log.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/domain-configuration/sparc-enterprise, SUNWdcsr
Interface Stability	Committed

関連項目

`svcs(1)`, `cfgadm_sbd(1M)`, `svcadm(1M)`, `svccfg(1M)`, `syslog(3C)`, `config_admin(3CFGADM)`, `libcfgadm(3LIB)`, `syslog.conf(4)`, `attributes(5)`, `smf(5)`, `dr(7d)`

注意事項

The `dcs` service is managed by the service management facility, `smf(5)`, under the fault management resource identifier (FMRI):

```
svc:/platform/sun4u/dcs:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前　dd – ファイルの変換とコピー

形式　/usr/bin/dd [*operand*=*value*]...

機能説明

dd は、指定した入力ファイルに可能な変換を行なって、指定した出力へコピーします。デフォルトでは、標準入力および標準出力が使用されます。raw 入出力装置の特性を利用するために入出力のブロックサイズで指定することが可能です。サイズは、バイト単位で指定し、数字のあとに k、b、または w を付加することができ、それぞれ 1024、512、または 2 の倍数として指定します。また、数字を x で区切ることにより乗算を表すことができます。

dd は、指定された入力ブロックサイズを用いて、入力データを 1 ブロックずつ読み込みます。その後、実際に渡されたデータブロックを処理します。そのサイズは、指定されたブロックサイズより小さい場合があります。dd は指定された変換処理をブロックに対して行い、結果のデータを、指定された出力ブロックサイズに従ってブロック単位で書き出します。

cbs は、ascii、asciib、unblock、ebcdic、ebcdicb、ibm、ibmb、または block 変換が指定されている場合にかぎり使用されます。最初の 2 つの指定では、cbs 文字は変換バッファーにコピーされ、任意の指定文字のマッピングが行われます。また後続の空白文字は切り捨てられ、行を送信する前に復帰改行が追加されます。残りの 3 つの指定では、復帰改行までの文字が変換バッファーに読み込まれ、サイズ cbs の出力レコードを構成するために空白文字が追加されます。ASCII ファイルは復帰改行文字を含むものとします。cbs が指定されていないかまたは 0 であると、ascii、asciib、ebcdic、ebcdicb、ibm、ibmb の各オプションは入力ファイルのブロック構造を変更せずに文字セットを変換します。unblock および block の各オプションは単純なファイルコピーを行います。

終了後に、dd は全体および部分的な入出力ブロック数を報告します。

オペランド　次のオペランドがサポートされています。

if=*file*　入力パス名。デフォルトは標準入力です。

of=*file*　出力パス名。デフォルトは標準出力です。seek=expr 変換が指定されていないとき、conv=notrunc も指定されていなければ、コピー処理の実行前に出力ファイルは切り捨てられます。seek=expr が指定され、conv=notrunc が指定されていなければ、コピー処理の結果としては dd がシークする出力ファイル中のブロックは保持されますが、出力ファイルのその他の部分は保持されません。シークするサイズと入力ファイルのサイズの合計が出力ファイルの元のサイズより小さい場合、コピー処理により出力ファイルは小さくなります。

ibs=*n*　入力ブロックサイズを *n* バイトとします。デフォルト値は 512 です。

obs=*n*
: 出力ブロックサイズを *n* バイトとします。デフォルト値は 512 です。

bs=*n*
: 入力ブロックサイズと出力ブロックサイズをともに *n* バイトとします。この設定は ibs= と obs= 設定よりも優先されます。sync、noerror、notrunc 以外の変換が 1 つも指定されない場合、各入力ブロックは複数の短いブロックを 1 つにまとめる処理は行われず、それぞれ単独のブロックとして出力側にコピーされます。

cbs=*n*
: block と unblock 用の変換ブロックサイズを *n* バイトに指定します。デフォルト値は 0 です。cbs= を指定しないかまたは 0 を指定した場合、block や unblock を使うと結果は不確定です。

このオプションは ASCII または EBCDIC 変換を指定した場合にだけ有効です。ascii と asciib オペランドを指定した場合、後続のスペース文字を消去する前に文字が ASCII に変換されるという点を除き、入力処理は unblock オペランド指定時と同じです。ebcdic、ebcdicb、ibm、または ibmb オペランドを指定した場合、末尾にスペース文字を追加した後で文字が EBCDIC または IBM EBCDICに変換されるという点を除き、入力処理は block オペランド指定時と同じです。

files=*n*
: 終了する前に、*n* 個の入力ファイルをコピーして連結します (入力が磁気テープまたは同様な装置の場合だけ有効です)。

skip=*n*
: コピーを開始する前に、指定された入力ブロックサイズを用いて *n* 個の入力ブロックを読み飛ばします。シーク可能なファイルに対しては、システムはそれらのブロックを読み込むか、あるいはシークを行います。シーク不可能なファイルに対しては、ブロックを読み込んで、そのデータを捨てます。

iseek=*n*
: コピーを行う前に、入力ファイルの先頭から *n* 個のブロックをシークします (skip の動作が遅いディスクファイルに適しています)。

oseek=*n*
: コピーを行う前に、出力ファイルの先頭から *n* 個のブロックをシークします。

seek=*n*
: コピーを行う前に、出力ファイルの先頭から *n* 個のブロックをスキップします (指定された出力ブロックサイズを使用)。シーク不可能なファイルに対しては、既存のブロックを読み込み、現在のファイルの終わり位置から指定

されたオフセット位置までの間にスペースがあれば、そのスペースをNULLバイトで埋めます。シーク可能なファイルに対しては、指定されたオフセット位置までをシークするか、またはシーク不可能なファイルの場合と同様にブロックを読み込みます。

`count=`*n*

n 個の入力ブロックだけをコピーします。

`conv=`*value*[`,`*value*. . .]

1つ以上の *value* をコンマで区切って記述します。各 value は次のいずれかです。

`ascii`

EBCDIC を ASCII に変換します。

`asciib`

BSD 互換の文字変換を使用して、EBCDIC を ASCII に変換します。

`ebcdic`

ASCII を EBCDIC に変換します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって `dd conv=unblock` でパイプラインを整えてください。

`ebcdicb`

BSD 互換の文字変換を使用して、ASCII を EBCDIC に変換します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって `dd conv=unblock` でパイプラインを整えてください。

`ibm`

ASCII から EBCDIC への変換とわずかに異なるマップを使用します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって `dd conv=unblock` でパイプラインを整えてください。

`ibmb`

BSD 互換の文字変換を使用して ASCII から EBCDIC への変換とわずかに異なるマップを使用します。復帰改行のない固定長の ASCII レコードを変換する場合は、前もって `dd conv=unblock` でパイプラインを整えてください。

`ascii` (または `asciib`)、`ebcdic` (または `ebcdicb`)、`ibm` (または `ibmb`) のうち 2 つ以上を同時に指定することはできません。

`block`

入力データを、入力ブロック境界に関係なく、復帰改行文字または `EOF` で終わる可変長レコードの集まりと見なします。各レコードは、変換ブロックサイズで指定した長さの固定

長レコードに変換されます。入力行に復帰改行文字があれば削除されます。変換ブロックサイズより短い行に関しては、ブロックを埋めるためにスペース文字が付加されます。変換ブロックサイズより長い行に関しては、ブロックサイズを満たす最大長になるように文字が捨てられます。切り捨てられた行の総数が報告されます。

`unblock` 固定長のレコードを可変長に変換します。変換ブロックサイズと等しい長さ分のバイト (それより短いデータしか残っていなければそのすべて) を読み込み、後続のスペース文字を削除して、復帰改行文字を付加します。

`block` と `unblock` を同時に指定することはできません。

`lcase` `LC_CTYPE` カテゴリ中のキーワードの `tolower` で指定された大文字を、対応する小文字にマップします。マッピングを指定されなかった文字は、この変換によって変更されることはありません。

`ucase` `LC_CTYPE` カテゴリ中のキーワードの `toupper` で指定された小文字を、対応する大文字にマップします。マッピングを指定されなかった文字は、この変換によって変更されることはありません。

`lcase` と `ucase` を同時に指定することはできません。

`swab` 入力データを 2 バイトずつの対になっているものとし、各々の対についてバイトの値を交換します。入力レコードの長さが奇数バイトの場合には、最終バイトは無視されます。

`noerror` 入力エラーが発生しても処理を停止しません。入力エラーが起こると、標準エラー出力に診断メッセージと、入力および出力ブロック数が出力されます。ブロック数の形式は、正常に終了したときに出力されるものと同じです。`sync` 変換が指定されたときは、入力データのうち得られなかったバイトは NULL バイトに置き換えられて通常どおりに処理されます。そうでない場合、入力ブロックは出力上には現れません。

`notrunc` 出力ファイルを切り捨てません。今回の `dd` 呼び出しで明示的に書き込まれなかったブロック

も出力ファイル中に保持します。(前述の of=*file* オペランドの説明を参照)。

sync　入力ブロックに NULL バイトを付加して、サイズが ibs= で指定した値に等しくなるようにします。なお block または unblock も指定されている場合には、NULL バイトの代わりにスペース文字を付加します。

conv= 以外のオペランドが複数回指定されたときは、最後に記述された operand=*value* が有効となります。

bs=、cbs=、ibs=、obs= に関しては、バイト単位でサイズを指定する式をアプリケーション側で提供しなければなりません。式 expr は次のいずれかです。

1. 正の 10 進数。
2. 正の 10 進数の後に 1024 倍を示す k を付加したもの。
3. 正の 10 進数の後に 512 倍を示す b を付加したもの。
4. 2 つ以上の正の 10 進数 (k や b を付加してもしなくてもよい) を文字 x で区切ったもの。その 2 つの値を乗算した結果を表します。

オペランドはすべて、入力データを読み込む前に処理されます。

使用法

ファイルが 2G バイト (2^{31} バイト) 以上ある場合の dd の動作については、largefile(5) を参照してください。

使用例

例 1　テープドライブ 0 から 1 へのコピー

次のコマンドは、標準的なデバイス命名規約を用いて、テープドライブ 0 から 1 へコピーします。

```
example% dd if=/dev/rmt/0h  of=/dev/rmt/1h
```

例 2　標準入力の先頭の 10 バイトの削除

次のコマンドは、標準入力の先頭の 10 バイトを削除します。

```
example% dd ibs=10  skip=1
```

例 3　テープを ASCII ファイルに読み込む

次のコマンドは、テープブロックごとに 10 個の 80 バイト EBCDIC カードイメージにブロック化した EBCDIC テープを ASCII ファイル x に読み込みます。

```
example% dd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase
```

例 4 テープの書き込みに conv=sync を使用

次のコマンドは、テープに書き込む場合に conv=sync を使用する例です。

```
example% tar cvf - . | compress | dd obs=1024k of=/dev/rmt/0 conv=sync
```

環境変数

dd の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。LANG、LC_ALL、LC_CTYPE、LC_MESSAGES、および NLSPATH。

終了ステータス

次の終了値が返されます。

0 ファイルは正常にコピーされました。

>0 エラーが発生しました。

noerror 変換が指定されていないときに入力エラーが発生すると、部分的に生成された出力ブロックがあればそれを出力ファイルに書き出し、診断メッセージを出力し、コピー処理を中止します。その他のエラーを検出した場合には、診断メッセージを出力し、コピー処理を中止します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実
標準	standards(5) を参照してください。

関連項目

cp(1), sed(1), tr(1), attributes(5), environ(5), largefile(5), standards(5)

診断

f+p records in(out) 読み取られた (書き込まれた) 完全 (f) および部分 (p) ブロック数

注意事項

ブロックサイズが異なるファイルシステム間でファイルのコピーを行う場合には、dd を使用しないでください。

ファイルをコピーするのにブロック型デバイスを使用すると、最終ブロックをブロック境界にあわせるために余分な NULL バイトがファイルに追加されます。

dd が ibs=X および obs=Y オペランドを用いてパイプから読み取ると、その出力は常にサイズ Y の固まりでブロック化されます。bs=Z が使用されると、出力ブロックはその時点でパイプから読み取ることができる大きさになります。

dd を使用してテープデバイスにファイルをコピーする場合、ファイルのサイズはデバイスのセクタサイズ (たとえば 512K) の倍数でなければなりません。任意のサイズのファイルをテープデバイスにコピーする場合は、tar(1) または cpio(1) を使用してください。

SIGINT が発生した場合、dd はステータス情報を標準エラー出力に書き出して処理を終了します。その他のシグナル発生時には、標準的な動作を行います。

名前

ddu – GUI ベースのデバイスドライバユーティリティー

形式

```
/usr/bin/ddu [--silent]
```

機能説明

GUI ベースのデバイスドライバユーティリティー ddu は、Oracle Solaris オペレーティングシステムを実行しているシステム上のデバイスに関する情報を提供します。ddu によって、Image Packaging System (IPS) に接続し、ドライバが接続されていないデバイスのデバイスドライバを検索できるようになります。そうすることで、ユーザーは不足しているドライバをインストールできるようになります。

ddu は、使いやすいインタフェースを備えた GUI です。このユーティリティーには、対応するテキストベースのユーティリティーである ddu-text(1M) があります。このテキストベースのユーティリティーについては、それ自体のマニュアルページで説明されています。

オプション

次のオプションがサポートされています。

`--silent`
: ddu は、サイレントに実行されます。不足しているドライバがあるデバイスが検出された場合は、通知が示されます。通知をクリックすると、DDU GUI が表示されます。

使用例

例 1　ddu GUI の起動

次のコマンドで、ddu GUI が起動されます。

```
% ddu
```

例 2　ddu のサイレントモードでの起動

次のコマンドで、ddu がサイレントモードで起動されます。

```
% ddu --silent
```

終了ステータス

`0`
: アプリケーションは正常に終了しました。

`>0`
: アプリケーションは、エラーが発生して終了しました。

ファイル

`/usr/bin/ddu`
: ddu の実行可能ファイルです。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	diagnostic/ddu
インタフェースの安定性	外部 (External)

その他の使用条件の属性値は、diagnostic/ddu/data、diagnostic/ddu/library、および diagnostic/ddu/locale です。

関連項目

ddu-text(1M), attributes(5)

名前

ddu-text – テキストベースのデバイスドライバユーティリティー

形式

```
ddu-text
```

機能説明

テキストベースのデバイスドライバユーティリティー ddu-text は、Oracle Solaris オペレーティングシステムを実行しているシステム上のデバイスに関する情報を提供します。ddu-text によって、Image Packaging System (IPS) に接続し、ドライバが接続されていないデバイスのデバイスドライバを検索できるようになります。そうすることで、ユーザーは不足しているドライバをインストールできるようになります。

ddu-text はテキストベースです。このユーティリティーには、対応する GUI である ddu(1M) があります。この GUI については、それ自体のマニュアルページで説明されています。

ddu-text には、コマンド行オプションはありません。

使用例

例 1　ddu-text の呼び出し

次のコマンドで、ddu-text が呼び出されます。

```
% ddu-text
```

終了ステータス

0
　アプリケーションは正常に終了しました。

>0
　アプリケーションは、エラーが発生して終了しました。

ファイル

/usr/bin/ddu-text
　ddu-text の実行可能ファイルです。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	diagnostic/ddu
インタフェースの安定性	外部 (External)

その他の使用条件の属性値は、diagnostic/ddu/data、diagnostic/ddu/library、および diagnostic/ddu/locale です。

関連項目

ddu(1M), attributes(5)

名前

devchassisd – devchassis daemon

形式

```
/usr/lib/devchassis/devchassisd [-d]
```

機能説明

The devchassis daemon, devchassisd, provides user-level services for the management of the /dev/chassis name space. It is a system daemon started by the Service Management Facility (see smf(5)). Its fault management resource identifier (FMRI) is:

```
svc:/system/devchassis:daemon
```

Note that devchassisd is a Consolidation Private interface. See attributes(5).

オプション

The following option is supported:

-d, --debug
: Run the daemon in standalone debug mode. Messages will be displayed on the controlling terminal instead of to syslog. And increased verbosity will be enabled to display more details about the internal operations of the daemon.

使用例

例 1 Enabling the devchassis Service

The following command enables the devchassis service:

```
# svcadm enable svc:/system/devchassis
```

例 2 Disabling the devchassis Service

The following command disables the devchassis service:

```
# svcadm disable svc:/system/devchassis
```

エラー

The devchassisd daemon uses syslog(3C) to report status and error messages. All of the messages are logged with the LOG_DAEMON facility. Error messages are logged with the LOG_ERR and LOG_NOTICE priorities, and informational messages are logged with the LOG_INFO priority. The default entries in the /etc/syslog.conf file log all of the devchassisd daemon error messages to the /var/adm/messages log.

ファイル

/usr/lib/devchassis/devchassisd
: devchassisd daemon binary.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Consolidation Private

関連項目

svcs(1), svcadm(1M), syslog(3C), syslog.conf(4), attributes(5), smf(5), devchassis(7FS)

注意事項

The devchassis service is managed by the service management facility, smf(5), under the fault management resource identifier (FMRI):

```
svc:/system/devchassis
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

The devchassis service is enabled by default.

名前

devfsadm, devfsadmd – administration command for /dev

形式

```
/usr/sbin/devfsadm [-C] [-c device_class] [-i driver_name]
     [ -n] [-r root_dir] [-s] [-t table_file] [-u] [-v]

/usr/lib/devfsadm/devfsadmd
```

機能説明

devfsadm maintains the /dev namespace. It replaces the previous suite of devfs administration tools including drvconfig(1M), disks(1M), tapes(1M), ports(1M), audlinks(1M), and devlinks(1M).

The default operation is to attempt to load every driver in the system and attach to all possible device instances. Next, devfsadm creates logical links to device nodes in /dev and /devices and loads the device policy.

devfsadmd(1M) is the daemon version of devfsadm(1M). The daemon is started during system startup and is responsible for handling both reconfiguration boot processing and updating /dev and /devices in response to dynamic reconfiguration event notifications from the kernel.

For compatibility purposes, drvconfig(1M), disks(1M), tapes(1M), ports(1M), audlinks(1M), and devlinks(1M) are implemented as links to devfsadm.

In addition to managing /dev, devfsadm also maintains the path_to_inst(4) database.

オプション

The following options are supported:

-C
: Cleanup mode. Prompt devfsadm to cleanup dangling /dev links that are not normally removed. If the -c option is also used, devfsadm only cleans up for the listed devices' classes.

-c *device_class*
: Restrict operations to devices of class *device_class*. Solaris defines the following values for *device_class*: disk, tape, port, audio, and pseudo. This option might be specified more than once to specify multiple device classes.

-i *driver_name*
: Configure only the devices for the named driver, *driver_name*.

-n
: Do not attempt to load drivers or add new nodes to the kernel device tree.

-s
: Suppress any changes to /dev. This is useful with the -v option for debugging.

-t *table_file*
: Read an alternate devlink.tab file. devfsadm normally reads /etc/devlink.tab.

`-u`
Activate and attach devices for drivers added with `add_drv(1M)` `-u`. Cannot be used together with `-n` or `-r`.

`-r` *root_dir*
Presume that the `/dev` directory trees are found under *root_dir*, not directly under `root` (`/`). No other use or assumptions are made about *root_dir*.

`-v`
Print changes to `/dev` in verbose mode.

終了ステータス

The following exit values are returned:

`0`
Successful completion.

`1`
An error occurred.

ファイル

`/devices`
device nodes directory

`/dev`
logical symbolic links to /devices

`/usr/lib/devfsadm/devfsadmd`
devfsadm daemon

`/dev/.devfsadm_dev.lock`
update lock file

`/dev/.devfsadm_daemon.lock`
daemon lock file

`/etc/security/device_policy`
device policy file

`/etc/security/extra_privs`
additional device privileges

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`svcs(1)`, `add_drv(1M)`, `modinfo(1M)`, `modload(1M)`, `modunload(1M)`, `rem_drv(1M)`, `svcadm(1M)`, `tapes(1M)`, `path_to_inst(4)`, `attributes(5)`, `privileges(5)`, `smf(5)`, `devfs(7FS)`

注意事項

This document does not constitute an API. The `/devices` directory might not exist or might have different contents or interpretations in a future release. The existence of this notice does not imply that any other documentation that lacks this notice constitutes an API.

`devfsadm` no longer manages the `/devices` name space. See `devfs(7FS)`.

As a daemon to support hot-plug and synchronous device naming, `devfsadm` is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/devfsadm:default
```

The status of the service can be queried using the `svcs(1)` command.

名前

device_allocate – enable and disable device allocation

形式

```
svc:/system/device/allocate:default
```

機能説明

The `device_allocate` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/device/allocate:default
```

An administrator with the Devices Security Rights Profile can enable or disable this service to provide for or remove the device allocation functionality as described in `allocate(1)`.

使用例

例 1 Enabling Device Allocation

The following command enables device allocation.

```
# svcadm enable svc:/system/device/allocate:default
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Volatile

関連項目

`allocate(1)`, `deallocate(1)`, `list_devices(1)`, `devfsadm(1M)`, `svcadm(1M)`, `attributes(5)`, `smf(5)`

名前

device_remap – administer the Solaris I/O remapping feature

形式

`/usr/platform/sun4v/sbin/device_remap [-v | -R dir]`

機能説明

Certain multi-node sun4v platforms, such as T5440 and T5240 servers, have an integrated PCI topology that cause the I/O device paths to change in a CPU node failover condition. The device remapping script, `device_remap`, remaps the device paths in `/etc/path_to_inst` file and the symlinks under `/dev` to match the hardware.

オプション

The following options are supported:

`-v`
: Displays the `/etc/path_to_inst` and `/dev` symlink changes.

`-R` *dir*
: Perform remapping on the `/etc/path_to_inst` and `/etc/path_to_inst` files in the root image at *dir*.

使用法

The primary function of `device_remap` is to remap the device paths in the `/etc/path_to_inst` file and the symlinks under `/dev` in a CPU node failover condition to match the hardware.

After adding CPU node(s) or removing CPU node(s), boot the system to the OBP prompt and use the following procedure:

1. Boot either the failsafe miniroot using: `boot -F failsafe` , or an install miniroot using `boot net -s` or similar command.
2. Mount the root disk as `/mnt`.
3. Change directory to the mounted root disk:

   ```
   # cd /mnt
   ```

4. Run `device_remap` script:

   ```
   # /mnt/usr/platform/sun4v/sbin/device_remap
   ```

5. Boot the system from disk.

All the error messages are self-explanatory, except for the error message “missing ioaliases node” which means the firmware on the system does not support device remapping.

使用例

例 1 Displaying Changes Following Failover

The following command displays the `path_to_inst` and `/dev` changes following a CPU node failover.

```
# device_remap -v
```

例 2 Changing Directory Prior to Any Changes

The following command changes the directory on which the boot image is mounted prior to making any changes.

```
# device_remap -R /newroot
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/platform
Interface Stability	Uncommitted

関連項目

boot(1M), attributes(5)

名前

devinfo – print device specific information

形式

`/usr/sbin/devinfo -i` *device*

`/usr/sbin/devinfo -p` *device*

機能説明

The `devinfo` command is used to print device specific information about disk devices on standard out. The command can only be used by the superuser.

オプション

`-i` Prints the following device information:

- Device name
- Software version (not supported and prints as `0`)
- Drive id number (not supported and prints as `0`)
- Device blocks per cylinder
- Device bytes per block
- Number of device partitions with a block size greater than zero

`-p` Prints the following device partition information:

- Device name
- Device major and minor numbers (in hexadecimal)
- Partition start block
- Number of blocks allocated to the partition
- Partition flag
- Partition tag

This command is used by various other commands to obtain device specific information for the making of file systems and determining partition information. If the device cannot be opened, an error message is reported.

オペランド

device Device name.

終了ステータス

`0` Successful operation.

`2` Operation failed.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`prtvtoc(1M)`, `attributes(5)`

名前

devlinks – adds /dev entries for miscellaneous devices and pseudo-devices

形式

`/usr/sbin/devlinks [-d] [-r rootdir] [-t table-file]`

機能説明

`devfsadm(1M)` is now the preferred command for `/dev` and `/devices` and should be used instead of `devlinks`.

`devlinks` creates symbolic links from the `/dev` directory tree to the actual block- and character-special device nodes under the `/devices` directory tree. The links are created according to specifications found in the *table-file* (by default `/etc/devlink.tab`).

`devlinks` is called each time the system is reconfiguration-booted, and can only be run after `drvconfig(1M)` is run.

The *table-file* (normally `/etc/devlink.tab`) is an ASCII file, with one line per record. Comment lines, which must contain a hash character ('#') as their first character, are allowed. Each entry must contain at least two fields, but may contain three fields. Fields are separated by single TAB characters.

The fields are:

devfs-spec

Specification of devinfo nodes that will have links created for them. This specification consists of one or more keyword-value pairs, where the keyword is separated from the value by an equal-sign ('='), and keyword-value pairs are separated from one another by semicolons.

The possible keywords are:

type

The devinfo device type. Possible values are specified in `ddi_create_minor_node(9F)`

name

The name of the node. This is the portion of the `/devices` tree entry name that occurs before the first '@' or ':' character.

addr[*n*]

The address portion of a node name. This is the portion of a node name that occurs between the '@' and the ':' characters. It is possible that a node may have a name without an address part, which is the case for many of the pseudo-device nodes. If a number is given after the *addr* it specifies a match of a particular comma-separated subfield of the address field: *addr1* matches the first subfield, *addr2* matches the second, and so on. *addr0* is the same as *addr* and matches the whole field.

minor[*n*]

The minor portion of a node name – the portion of the name after the ':'. As with *addr* above, a number after the *minor* keyword specifies a subfield to match.

Of these four specifications, only the *type* specification must always be present.

name

Specification of the `/dev` links that correspond to the devinfo nodes. This field allows `devlinks` to determine matching `/dev` names for the `/devices` nodes it has found. The specification of this field uses escape-sequences to allow portions of the `/devices` name to be included in the `/dev` name, *or* to allow a counter to be used in creating node names. If a counter is used to create a name, the portion of the name before the counter must be specified absolutely, and all names in the `/dev/`-subdirectory that match (up to and including the counter) are considered to be subdevices of the same device. This means that they should all point to the same directory, `name` and `address` under the `/devices/`-tree

The possible escape-sequences are:

`\D` Substitute the `device-name` (name) portion of the corresponding devinfo node-name.

`\A`*n* Substitute the *n*th component of the `address` component of the corresponding devinfo node name. Sub-components are separated by commas, and sub-component `0` is the whole `address` component.

`\M`*n* Substitute the *n*th sub-component of the `minor` component of the corresponding devinfo node name. Sub-components are separated by commas, and sub-component `0` is the whole `minor` component.

`\N`*n* Substitute the value of a 'counter' starting at *n*. There can be only one counter for each dev-spec, and counter-values will be selected so they are as low as possible while not colliding with already-existing link names.

In a dev-spec the `counter` sequence should not be followed by a digit, either explicitly or as a result of another escape-sequence expansion. If this occurs, it would not be possible to correctly match already-existing links to their counter entries, since it would not be possible to unambiguously parse the already-existing `/dev`-name.

extra-dev-link

Optional specification of an extra `/dev` link that points to the initial `/dev` link (specified in field 2). This field may contain a `counter` escape-sequence (as described for the *dev-spec* field) but may not contain any of the other escape-sequences. It provides a way to specify an alias of a particular `/dev` name.

オプション

The following options are supported:

`-d`	Debugging mode – print out all `devinfo` nodes found, and indicate what links would be created, but do not do anything.
`-r` *rootdir*	Use *rootdir* as the root of the `/dev` and `/devices` directories under which the device nodes and links are created. Changing the root directory does not change the location of the `/etc/devlink.tab` default table, nor is the root directory applied to the filename supplied to the `-t` option.
`-t` *table-file*	Set the table file used by `devlinks` to specify the links that must be created. If this option is not given, `/etc/devlink.tab` is used. This option gives a way to instruct `devlinks` just to perform a particular piece of work, since just the links-types that `devlinks` is supposed to create can be specified in a command-file and fed to `devlinks`.

エラー

If `devlinks` finds an error in a line of the *table-file* it prints a warning message on its standard output and goes on to the next line in the *table-file* without performing any of the actions specified by the erroneous rule.

If it cannot create a link for some filesystem-related reason it prints an error-message and continues with the current rule.

If it cannot read necessary data it prints an error message and continues with the next *table-file* line.

使用例

例 1　Using the /etc/devlink.tab Fields

The following are examples of the `/etc/devlink.tab` fields:

```
type=pseudo;name=win    win\M0
type=ddi_display    framebuffer/\M0    fb\N0
```

The first example states that all devices of type `pseudo` with a name component of `win` will be linked to `/dev/win`*x*, where *x* is the minor-component of the *devinfo-name* (this is always a single-digit number for the `win` driver).

The second example states that all devinfo nodes of type `ddi_display` will be linked to entries under the `/dev/framebuffer` directory, with names identical to the entire minor component of the `/devices` name. In addition an extra link will be created pointing from `/dev/fb`*n* to the entry under `/dev/framebuffer`. This entry will use a counter to end the name.

ファイル

`/dev`	entries for the miscellaneous devices for general use
`/devices`	device nodes
`/etc/devlink.tab`	the default rule-file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

devfsadm(1M), attributes(5), devfs(7FS), ddi_create_minor_node(9F)

使用上の留意点

It is very easy to construct mutually-contradictory link specifications, or specifications that can never be matched. The program does not check for these conditions.

名前

devnm – device name

形式

`/usr/sbin/devnm` *name* [*name*]...

機能説明

The `devnm` command identifies the special file associated with the mounted file system where the argument *name* resides. One or more *name* can be specified.

使用例

例 1 Using the devnm Command

Assuming that `/usr` is mounted on `/dev/dsk/c0t3d0s6`, the following command :

```
/usr/sbin/devnm /usr
```

produces:

```
/dev/dsk/c0t3d0s6 /usr
```

ファイル

`/dev/dsk/*`

`/etc/mnttab`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`mnttab(4)`, `attributes(5)`

名前

devprop – display device properties

形式

`/usr/sbin/devprop [-n` *device-path*`] [-vq] [-{b|i|l|s}] [`*property*`[...]]`

機能説明

The `devprop` command displays named device properties from the device tree.

If a device path is specified on the command line, `devprop` displays device properties for that device node.

The output formats, while Volatile, are patterned after those of `prtconf(1M)`.

オプション

The following options are supported:

`-b`
: The properties to be output are sequences of bytes (`DI_PROP_TYPE_BYTES`). `-b`, `-i`, `-l`, and `-s` are mutually exclusive of one another.

`-i`
: The properties to be output are integers (`DI_PROP_TYPE_INT`).

`-l`
: The properties to be output are 64–bit integers (`DI_PROP_TYPE_INT64`).

`-s`
: The properties to be output are strings (`DI_PROP_TYPE_STRING`) (the default).

`-n` *device-path*
: The path to a target device node for which properties are displayed. The default path is that of the root node (equivalent to specifying `-n /`).

`-q`
: Specifies quoted output mode, in which string properties are output surrounded by double quotation marks (").

`-v`
: Specifies verbose mode, in which the name of the property is output before its value.

オペランド

The following operand is supported:

property...
: Name of the property to be displayed.

終了ステータス

`0`
: No error occurred.

`non-zero`
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

The command invocation and output are both Volatile. The location of the utility is Committed.

関連項目

prtconf(1M), libdevinfo(3LIB), attributes(5)

名前

df – 使用可能なディスクブロックおよび使用可能なファイル数の表示

形式

```
df [-F FSType] [-abeghklntPVvZ]
     [-o FSType-specific_options]
     [block_device | directory | file | resource ...]
```

機能説明

df コマンドは、マウントされているもしくはマウントされていないファイルシステムが使用しているディスク容量を表示します。さらに、使用中の容量、使用可能な容量、ファイルシステムの全容量のうちどの程度が使用されたかを表示します。対象のファイルシステムとしては、デバイス、または特定のファイルシステム上のファイルあるいはディレクトリを指定します。

オペランドもオプションも指定せずに df を実行すると、全ファイルシステムに関する情報が出力されます。

df は一部の *FSTypes* (ファイルシステムタイプ) でサポートされない場合があります。

自動マウンタがまだマウントしていないネットワーク上のマウントポイントで df を実行する場合、ファイルシステムサイズの情報がゼロとして出力されます。自動マウンタがファイルシステムをマウントすると、サイズの情報は正確に出力されるようになります。

オプション

次のオプションがサポートされています。

-a
: /etc/mnttab のエントリ (mnttab(4) を参照) に ignore オプションセットを含むファイルシステムについても情報を出力します。

-b
: 使用可能な容量の合計を、K バイト単位で報告します。

-e
: 使用可能なファイル数だけを報告します。

-F *FSType*
: df コマンドが動作する *FSType* を指定します。-F オプションはマウントされていないファイルシステムを使用するためのオプションです。*FSType* は、このオプションで指定するか、あるいは /etc/vfstab 内の記述(vfstab(4) を参照) から特定できるようにしておく必要があります。後者の場合、具体的には *directory*、*block_device*、または *resource* 引数の指定値とテーブル中のエントリの値を一致させるか、あるいは /etc/default/fs を参照して決定されます。詳細は default_fs(4) を参照してください。

-g
: statvfs(2) 構造体全体を報告します。このオプションはマウントされているファイルシステムに対してのみ有効です。-o オプションと同時に指定することはできません。このオプションは、-b、-e、-k、-n、-P、-t の各オプション (指定されている場合) よりも優先されます。

-h
-k と似ていますが、サイズ情報がより読みやすい形式で表示されます。1 つのファイルシステムにつき 1 行の情報が出力されます。情報の内容は、ファイルシステム名、そのファイルシステムに割り当てられている容量の合計、既存のファイルに割り当てられている容量の合計、特権を持たないユーザーが新たなファイルを生成する場合に使用できる容量の合計、そのファイルシステム上の全ファイルに現在割り当てられている通常使用可能な容量の割合 (パーセント単位) です。すべてのサイズは縮小され、読みやすい形式で表示されます。たとえば、14K、234M、2.7G、3.0T などのようになります。縮尺は、1024 を除数として行われます。

このオプションは -b、-e、-g、-k、-n、-t、および -V オプションを無効にします。このオプションはマウントされているファイルシステム上でのみ有効であり、-o オプションと一緒には使用できません。

-k
割り当てられているディスク容量を K バイト単位で出力します。1 つのファイルシステムにつき 1 行の情報が出力されます。情報の内容は、ファイルシステム名、そのファイルシステムに割り当てられている容量の合計、既存のファイルに割り当てられている容量の合計、特権を持たないユーザーが新たなファイルを生成する場合に使用できる容量の合計、そのファイルシステム上の全ファイルに現在割り当てられている通常使用可能な容量の割合 (パーセント単位) です。このオプションは、-b、-e、-n、および -t オプションを無効にします。また、-v オプションと同時に指定できません。

-l
ローカルファイルシステムについての情報だけを報告します。このオプションはマウントされているファイルシステムに対してのみ有効です。-o オプションと同時に指定することはできません。

-n
FSType の名前だけを報告します。オペランド指定を省略すると、このオプションはマウントされているファイルシステムタイプの一覧を出力します。このオプションはマウントされているファイルシステムに対してのみ有効です。-o オプションと同時に指定することはできません。

-o *FSType-specific_options*
FSType 固有のオプションを指定します。オプションとオプションとの間はコンマだけで区切り、スペースは入れないでください。詳細については *FSType* コマンド用のマニュアルページを参照してください。

-t
合計値を含む完全なリストを出力します。このオプションは、-b、-e、-n の各オプション (指定されている場合) よりも優先されます。

-P
-h オプションと同じですが、出力の単位は 512 バイトとなります。

`-V`
指定されたコマンド行の全内容のエコーだけを行い、コマンド自体は実行しません。コマンド行の内容としては、ユーザーが指定したオプションやオペランドに加え、`/etc/mnttab`、`/etc/vfstab`、`/etc/default/fs` の各ファイルから得られた情報が付加されます。このオプションは、コマンド行を確認および検証するときに使用します。

`-v`
`-k` オプションと似ていますが、サイズ情報がそれぞれのファイルシステムを構成する最小ブロックサイズの倍数で表示されます。

`-k` オプションと同時に指定できません。

1 つのファイルシステムにつき 1 行の情報が出力されます。1 行の情報の内容は次のとおりです。

- ファイルシステムのマウントポイント
- ファイルシステム名
- そのファイルシステムに割り当てられているブロックの合計数
- 既存のファイルに割り当てられているブロック数
- 特権を持たないユーザーが新たなファイルを生成する場合に使用できるブロック数
- ファイルに使用されているブロックの割合 (パーセント単位)

`-Z`
すべての可視ゾーン内にあるマウントを表示します。デフォルトでは、`df` は、現在のゾーン内にあるマウントだけを表示します。このオプションは、非大域ゾーンでは効果がありません。

オペランド

`df` ユーティリティーは、*block_device*、*directory*、*file*、*resource* の優先度に従ってオペランドを解釈します。次のオペランドがサポートされています。

block_device
ブロック型特殊デバイス (たとえば、`/dev/dsk/c1d0s7`) を表します。

directory
有効なディレクトリ名を指定します。`df` は *directory* を含むファイルシステムについて報告します。

file
有効なファイル名を指定します。`df` は *file* があるファイルシステムについて報告します。

resource
NFS リソース名を指定します。

使用法

ファイルが 2G バイト (2^{31} バイト) 以上ある場合の df の動作については、largefile(5)を参照してください。

使用例

例 1　df コマンドを実行する

以下は、df コマンドとその出力の例です。

```
example% /usr/bin/df

/                  (/dev/dsk/c0t0d0s0 ):  287530 blocks    92028 files
/system/contract   (ctfs              ):       0 blocks 2147483572 files
/system/object     (objfs             ):       0 blocks 2147483511 files
/usr               (/dev/dsk/c0t0d0s6 ): 1020214 blocks   268550 files
/proc              (/proc             ):       0 blocks      878 files
/dev/fd            (fd                ):       0 blocks        0 files
/etc/mnttab        (mnttab            ):       0 blocks        0 files
/var/run           (swap              ):  396016 blocks     9375 files
/tmp               (swap              ):  396016 blocks     9375 files
/opt               (/dev/dsk/c0t0d0s5 ):  381552 blocks    96649 files
/export/home       (/dev/dsk/c0t0d0s7 ):  434364 blocks   108220 files
```

各列は、左から、マウントポイント、デバイス (または、df -k による「ファイルシステム」)、空きブロック、および空きファイルを示します。contract ファイルシステムの場合、マウントポイントは /system/contract、ファイルシステムは ctfs (SMF が使用)、空きブロックは 0、そして、空きファイルは 2147483582 (INTMAX-1) です。object ファイルシステムの場合、マウントポイントは /system/object、ファイルシステムは objfs (objfs(7FS) を参照)、空きブロックは 0、そして、空きファイルは 2147483511 です。

例 2　ファイルシステム /usr に関する情報を出力する

次の例は、ファイルシステム /usr に関する情報を出力します。

```
example% /usr/bin/df -P /usr
```

例 3　ファイルシステム /usr に関する情報を出力する (/usr/src が /usr ファイルシステムの一部である場合)

/usr/src が /usr ファイルシステムの一部である場合、次の例も前述の例と同じ結果を出力します。

```
example% /usr/bin/df -P /usr/src
```

例 4　すべての ufs ファイルシステムの i ノード使用率を表示する

次の例は、すべての ufs ファイルシステムにおける i ノードの使用率を表示します。

```
example%/usr/bin/df -F ufs -o i
```

環境変数

設定した場合、通常ファイルを表示するヘッダーは、すべてノードを表示するようになります。df の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。 LANG、LC_ALL、LC_CTYPE、LC_MESSAGES、および NLSPATH。

終了ステータス

次の終了値が返されます。

0
: 正常終了。

>0
: エラーが発生しました。

ファイル

/dev/dsk/*
: ディスクデバイス

/etc/default/fs
: デフォルトのローカルのファイルシステムタイプ。デフォルト値は、/etc/default/fs 内で次のように設定されています。たとえば、次のように指定します。 FSType が指定されていない場合に LOCAL がコマンドに対するデフォルトパーティションの場合は、LOCAL=ufs と指定します。

/etc/mnttab
: マウントテーブル

/etc/vfstab
: 各ファイルシステム用のデフォルトのパラメータリスト

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実
標準	standards(5) を参照してください。

関連項目

find(1), df_ufs(1M), mount(1M), statvfs(2), default_fs(4), mnttab(4), vfstab(4), attributes(5), environ(5), largefile(5), standards(5), objfs(7FS)

注意事項

ファイルシステムで UFS ログが有効になっている場合、ログで使用されるディスク容量は df の報告に反映されます。ログはファイルシステム上の空きブロックから割り当てられ、ファイルシステムの 1G バイトあたり約 1M バイト、最大 256M バイトのサイズになります。ログサイズは、ファイルシステム内に存在するシリンダグループの数に応じて、さらに大きくなることがあります (最大 512M バイト)。

名前

dfmounts – display mounted resource information

形式

```
dfmounts [-F FSType] [-h] [-o specific_options]
      [restriction]...
```

機能説明

`dfmounts` shows the local resources shared through a distributed file system *FSType* along with a list of clients that have the resource mounted. If *restriction* is not specified, `dfmounts` shows file systems that are currently shared on any NFS server. *specific_options* as well as the availability and semantics of *restriction* are specific to particular distributed file system types.

If `dfmounts` is entered without arguments, remote resources currently mounted on the local system are displayed, regardless of file system type. However, the `dfmounts` command does not display the names of NFS Version 4 clients.

dfmounts Output

The output of `dfmounts` consists of an optional header line (suppressed with the `-h` flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:

resource server pathname clients ...

where:

resource	Specifies the resource name that must be given to the `mount(1M)` command.
server	Specifies the system from which the resource was mounted.
pathname	Specifies the pathname that must be given to the `share(1M)` command.
clients	Is a comma-separated list of systems that have mounted the resource. Clients are listed in the form *domain.*, *domain.system*, or *system*, depending on the file system type.

A field can be null. Each null field is indicated by a hyphen (–) unless the remainder of the fields on the line are also null, in which case the hyphen can be omitted.

Fields with whitespace are enclosed in quotation marks (" ").

オプション

`-F` *FSType*	Specify filesystem type. Defaults to the first entry in `/etc/dfs/fstypes`. `Note`: currently the only valid *FSType* is `nfs`.
`-h`	Suppress header line in output.
`-o` *specific_options*	Specify options specific to the filesystem provided by the `-F` option. `Note`: currently no options are supported.

ファイル

`/etc/dfs/fstypes`	file system types

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

dfshares(1M), mount(1M), share(1M), unshare(1M), attributes(5)

名前

dfmounts_nfs – display mounted NFS resource information

形式

`dfmounts [-F nfs] [-h] [server]...`

機能説明

`dfmounts` shows the local resources shared through NFS, along with the list of clients that have mounted the resource. The `-F` flag may be omitted if NFS is the only file system type listed in the file `/etc/dfs/fstypes`.

`dfmounts` without options, displays all remote resources mounted on the local system, regardless of file system type.

The output of `dfmounts` consists of an optional header line (suppressed with the `-h` flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:

resource server pathname clients ...

where

resource Does not apply to NFS. Printed as a hyphen (-).

server Specifies the system from which the resource was mounted.

pathname Specifies the pathname that must be given to the `share(1M)` command.

clients Is a comma-separated list of systems that have mounted the resource.

オプション

`-F nfs` Specifies the `nfs`-FSType.

`-h` Suppress header line in output.

server Displays information about the resources mounted from each server, where *server* can be any system on the network. If no server is specified, the *server* is assumed to be the local system.

ファイル

`/etc/dfs/fstypes`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目

`mount(1M)`, `share(1M)`, `unshare(1M)`, `attributes(5)`

名前

dfshares – list available resources from remote or local systems

形式

`dfshares [-F FSType] [-h] [-o specific_options] [server]...`

機能説明

`dfshares` provides information about resources available to the host through a distributed file system of type *FSType*. *specific_options* as well as the semantics of *server* are specific to particular distributed file systems.

If `dfshares` is entered without arguments, all resources currently shared on the local system are displayed, regardless of file system type.

The output of `dfshares` consists of an optional header line (suppressed with the `-h` flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:

resource server access transport

where

resource Specifies the resource name that must be given to the `mount(1M)` command.

server Specifies the name of the system that is making the resource available.

access Specifies the access permissions granted to the client systems, either `ro` (for read-only) or `rw` (for read/write). If `dfshares` cannot determine access permissions, a hyphen (–) is displayed.

transport Specifies the transport provider over which the resource is shared.

A field may be null. Each null field is indicated by a hyphen (–) unless the remainder of the fields on the line are also null; in which case, the hyphen may be omitted.

オプション

`-F` *FSType* Specify filesystem type. Defaults to the first entry in `/etc/dfs/fstypes`.

`-h` Suppress header line in output.

`-o` *specific_options* Specify options specific to the filesystem provided by the `-F` option.

ファイル

`/etc/dfs/fstypes`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`dfmounts(1M)`, `mount(1M)`, `share(1M)`, `unshare(1M)`, `attributes(5)`

名前

dfshares_nfs – list available NFS resources from remote systems

形式

`dfshares [-F nfs] [-h] [server]...`

機能説明

`dfshares` provides information about resources available to the host through NFS. The `-F` flag may be omitted if NFS is the first file system type listed in the file `/etc/dfs/fstypes`.

The query may be restricted to the output of resources available from one or more servers.

`dfshares` without arguments displays all resources shared on the local system, regardless of file system type.

Specifying *server* displays information about the resources shared by each server. *Server* can be any system on the network. If no server is specified, then *server* is assumed to be the local system.

The output of `dfshares` consists of an optional header line (suppressed with the `-h` flag) followed by a list of lines containing whitespace-separated fields. For each resource, the fields are:

resource server access transport

where

resource Specifies the resource name that must be given to the `mount(1M)` command.

server Specifies the system that is making the resource available.

access Specifies the access permissions granted to the client systems; however, `dfshares` cannot determine this information for an NFS resource and populates the field with a hyphen (-).

transport Specifies the transport provider over which the *resource* is shared; however, `dfshares` cannot determine this information for an NFS resource and populates the field with a hyphen (-).

A field may be null. Each null field is indicated by a hyphen (-) unless the remainder of the fields on the line are also null; in which case, the hyphen may be omitted.

オプション

`-F nfs` Specify the NFS file system type

`-h` Suppress header line in output.

ファイル

`/etc/dfs/fstypes`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目 mount(1M), share(1M), unshare(1M), attributes(5)

名前

df_ufs – report free disk space on ufs file systems

形式

`df -F ufs [generic_options] [-o i] [directory | special]`

機能説明

`df` displays the amount of disk space occupied by `ufs` file systems, the amount of used and available space, and how much of the file system's total capacity has been used.The amount of space reported as used and available is less than the amount of space in the file system; this is because the system reserves a fraction of the space in the file system to allow its file system allocation routines to work well. The amount reserved is typically about 10%; this can be adjusted using `tunefs(1M)`. When all the space on the file system except for this reserve is in use, only the superuser can allocate new files and data blocks to existing files. When the file system is overallocated in this way, `df` might report that the file system is more than 100% utilized.If neither *directory* nor *special* is specified, `df` displays information for all mounted ufs file systems.

オプション

The following options are supported:

generic_options Options supported by the generic `df` command. See `df(1M)` for a description of these options.

`-o` Specify `ufs` file system specific options. The available option is:

`i` Report the number of used and free inodes. This option can not be used with *generic_options*.

ファイル

`/etc/mnttab` list of file systems currently mounted

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os, system/xopen/xcu4

関連項目

`df(1M)`, `fsck(1M)`, `fstyp(1M)`, `tunefs(1M)`, `mnttab(4)`, `attributes(5)`, `ufs(7FS)`,

注意事項

`df` calculates its results differently for mounted and unmounted file systems. For unmounted systems, the numbers reflect the 10% reservation. This reservation is not reflected in `df` output for mounted file systems. For this reason, the available space reported by the generic command can differ from the available space reported by this module.

`df` might report remaining capacity even though `syslog` warns `filesystem full`. This issue can occur because `df` only uses the available fragment count to calculate available space, but the file system requires contiguous sets of fragments for most allocations.

If you suspect that you have exhausted contiguous fragments on your file system, you can use the `fstyp(1M)` utility with the `-v` option. In the `fstyp` output, look at the `nbfree` (number of blocks free) and `nffree` (number of fragments free) fields. On unmounted filesystems, you can use `fsck(1M)` and observe the last line of output, which reports, among other items, the number of fragments and the degree of fragmentation. See `fsck(1M)`.

名前

dhcpagent – Dynamic Host Configuration Protocol (DHCP) client daemon

形式

```
dhcpagent [-a] [ -d n] [-f] [-v]
```

機能説明

dhcpagent implements the client half of the Dynamic Host Configuration Protocol (DHCP) for machines running Solaris software.

The dhcpagent daemon obtains configuration parameters for the client (local) machine's network interfaces from a DHCP server. These parameters may include a lease on an IP address, which gives the client machine use of the address for the period of the lease, which may be infinite. If the client wishes to use the IP address for a period longer than the lease, it must negotiate an extension using DHCP. For this reason, dhcpagent must run as a daemon, terminating only when the client machine powers down.

For IPv4, the dhcpagent daemon is controlled through ifconfig(1M).

dhcpagent can be invoked as a user process, albeit one requiring root privileges, but this is not necessary, as ifconfig(1M) will start it automatically.

For IPv6, the dhcpagent daemon is invoked automatically by in.ndpd(1M). It can also be controlled through ifconfig(1M), if necessary.

When invoked, dhcpagent enters a passive state while it awaits instructions from ifconfig(1M) or in.ndpd(1M). When it receives a command to configure an interface, it brings up the interface (if necessary) and starts DHCP. Once DHCP is complete, dhcpagent can be queried for the values of the various network parameters. In addition, if DHCP was used to obtain a lease on an address for an interface, it configures the address for use. When a lease is obtained, it is automatically renewed as necessary. If the lease cannot be renewed, dhcpagent will unconfigure the address, but the interface will be left up and dhcpagent will attempt to acquire a new address lease. dhcpagent monitors system suspend/resume events and will validate any non-permanent leases with the DHCP server upon resume. Similarly, dhcpagent monitors link up/down events and will validate any non-permanent leases with the DHCP server when the downed link is brought back up. The lease validation mechanism will restart DHCP if the server indicates that the existing lease is no longer valid. If the server cannot be contacted, then the existing lease will continue. This behavior can be modified with the VERIFIED_LEASE_ONLY parameter in the /etc/default/dhcpagent file. See the description of this parameter below.

For IPv4, if the configured interface is found to be unplumbed, or to have a different IP address, subnet mask, or broadcast address from those obtained from DHCP, the interface is abandoned from DHCP control.

For IPv6, dhcpagent automatically plumbs and unplumbs logical interfaces as necessary for the IPv6 addresses supplied by the server. The IPv6 prefix length (netmask) is not set by the DHCPv6 protocol, but is instead set by in.ndpd(1M) using prefix information obtained by Router Advertisements. If any of the logical interfaces created by dhcpagent is unplumbed, or

configured with a different IP address, it will be abandoned from DHCP control. If the link-local interface is unplumbed, then all addresses configured by DHCP on that physical interface will be removed.

In addition to DHCP, `dhcpagent` also supports BOOTP (IPv4 only). See RFC 951, Bootstrap Protocol. Configuration parameters obtained from a BOOTP server are treated identically to those received from a DHCP server, except that the IP address received from a BOOTP server always has an infinite lease.

DHCP also acts as a mechanism to configure other information needed by the client, for example, the domain name and addresses of routers. Aside from the IP address, and for IPv4 alone, the netmask, broadcast address, and default router, the agent does not directly configure the workstation, but instead acts as a database which may be interrogated by other programs, and in particular by `dhcpinfo(1)`.

On clients with a single interface, this is quite straightforward. Clients with multiple interfaces may present difficulties, as it is possible that some information arriving on different interfaces may need to be merged, or may be inconsistent. Furthermore, the configuration of the interfaces is asynchronous, so requests may arrive while some or all of the interfaces are still unconfigured. To handle these cases, one interface may be designated as primary, which makes it the authoritative source for the values of DHCP parameters in the case where no specific interface is requested. See `dhcpinfo(1)` and `ifconfig(1M)` for details.

All DHCP packets sent by `dhcpagent` include a vendor class identifier (RFC 2132, option code 60; RFC 3315, option code 16). This identifier is the same as the platform name returned by the `uname -i` command, except:

- Any commas in the platform name are changed to periods.
- If the name does not start with a stock symbol and a comma, it is automatically prefixed with `SUNW`.

Messages

The `dhcpagent` daemon writes information and error messages in five categories:

critical
: Critical messages indicate severe conditions that prevent proper operation.

errors
: Error messages are important, sometimes unrecoverable events due to resource exhaustion and other unexpected failure of system calls; ignoring errors may lead to degraded functionality.

warnings
: Warnings indicate less severe problems, and in most cases, describe unusual or incorrect datagrams received from servers, or requests for service that cannot be provided.

informational
: Informational messages provide key pieces of information that can be useful to debugging a DHCP configuration at a site. Informational messages are generally controlled by the `-v` option. However, certain critical pieces of information, such as the IP address obtained, are always provided.

debug
: Debugging messages, which may be generated at two different levels of verbosity, are chiefly of benefit to persons having access to source code, but may be useful as well in debugging difficult DHCP configuration problems. Debugging messages are only generated when using the `-d` option.

When `dhcpagent` is run without the `-f` option, all messages are sent to the system logger `syslog(3C)` at the appropriate matching priority and with a facility identifier `LOG_DAEMON`. When `dhcpagent` is run with the `-f` option, all messages are directed to standard error.

DHCP Events and User-Defined Actions

If an executable (binary or script) is placed at `/etc/dhcp/eventhook`, the `dhcpagent` daemon will automatically run that program when any of the following events occur:

`BOUND` and `BOUND6`
: These events occur during interface configuration. The event program is invoked when `dhcpagent` receives the DHCPv4 ACK or DHCPv6 Reply message from the DHCP server for the lease request of an address, indicating successful initial configuration of the interface. (See also the `INFORM` and `INFORM6` events, which occur when configuration parameters are obtained without address leases.)

`EXTEND` and `EXTEND6`
: These events occur during lease extension. The event program is invoked just after `dhcpagent` receives the DHCPv4 ACK or DHCPv6 Reply from the DHCP server for the DHCPv4 REQUEST (renew) message or the DHCPv6 Renew or Rebind message.

  Note that with DHCPv6, the server might choose to remove some addresses, add new address leases, and ignore (allow to expire) still other addresses in a given Reply message. The `EXTEND6` event occurs when a Reply is received that leaves one or more address leases still valid, even if the Reply message does not extend the lease for any address. The event program is invoked just before any addresses are removed, but just after any new addresses are added. Those to be removed will be marked with the `IFF_DEPRECATED` flag.

`EXPIRE` and `EXPIRE6`
: These events occur during lease expiration. For DHCPv4, the event program is invoked just before the leased address is removed from an interface. For DHCPv6, the event program is invoked just before the last remaining leased addresses are removed from the interface.

`DROP` and `DROP6`
: These events occur during the period when an interface is dropped. The event program is invoked just before the interface is removed from DHCP control. If the interface has been

abandoned due the user unplumbing the interface, then this event will occur after the user's action has taken place. The interface might not be present.

`INFORM` and `INFORM6`
: These events occur when an interface acquires new or updated configuration information from a DHCP server by means of the DHCPv4 `INFORM` or the DHCPv6 Information-Request message. These messages are sent using an `ifconfig(1M)` `dhcp inform` command or when the DHCPv6 Router Advertisement `O` (letter 0) bit is set and the `M` bit is not set. Thus, these events occur when the DHCP client does not obtain an IP address lease from the server, and instead obtains only configuration parameters.

`LOSS6`
: This event occurs during lease expiration when one or more valid leases still remain. The event program is invoked just before expired addresses are removed. Those being removed will be marked with the `IFF_DEPRECATED` flag.

  Note that this event is not associated with the receipt of the Reply message, which occurs only when one or more valid leases remain, and occurs only with DHCPv6. If all leases have expired, then the EXPIRE6 event occurs instead.

`RELEASE` and `RELEASE6`
: This event occurs during the period when a leased address is released. The event program is invoked just before `dhcpagent` relinquishes the address on an interface and sends the DHCPv4 `RELEASE` or DHCPv6 Release packet to the DHCP server.

The system does not provide a default event program. The file `/etc/dhcp/eventhook` is expected to be owned by root and have a mode of 755.

The event program will be passed two arguments, the interface name and the event name, respectively. For DHCPv6, the interface name is the name of the physical interface.

The event program can use the `dhcpinfo(1)` utility to fetch additional information about the interface. While the event program is invoked on every event defined above, it can ignore those events in which it is not interested. The event program runs with the same privileges and environment as `dhcpagent` itself, except that `stdin`, `stdout`, and `stderr` are redirected to `/dev/null`. Note that this means that the event program runs with root privileges.

If an invocation of the event program does not exit after 55 seconds, it is sent a `SIGTERM` signal. If does not exit within the next three seconds, it is terminated by a `SIGKILL` signal.

See EXAMPLES for an example event program.

オプション

The following options are supported:

`-a`
: Adopt a configured IPv4 interface. This option is for use with diskless DHCP clients. In the case of diskless DHCP, DHCP has already been performed on the network interface

providing the operating system image prior to running `dhcpagent`. This option instructs the agent to take over control of the interface. It is intended primarily for use in boot scripts.

The effect of this option depends on whether the interface is being adopted.

If the interface is being adopted, the following conditions apply:

`dhcpagent` uses the client id specified in `/chosen`:<*client_id*>, as published by the PROM or as specified on a `boot(1M)` command line. If this value is not present, the client id is undefined. The DHCP server then determines what to use as a client id. It is an error condition if the interface is an Infiniband interface and the PROM value is not present.

If the interface is not being adopted:

`dhcpagent` uses the value stored in `/etc/default/dhcpagent`. If this value is not present, the client id is undefined. If the interface is Infiniband and there is no value in `/etc/default/dhcpagent`, a client id is generated as described by the draft document on DHCP over Infiniband, available at:

`http://www.ietf.org`

`-d` *n*
Set debug level to *n*. Two levels of debugging are currently available, 1 and 2; the latter is more verbose.

`-f`
Run in the foreground instead of as a daemon process. When this option is used, messages are sent to standard error instead of to `syslog(3C)`.

`-v`
Provide verbose output useful for debugging site configuration problems.

使用例

例 1 Example Event Program

The following script is stored in the file `/etc/dhcp/eventhook`, owned by root with a mode of 755. It is invoked upon the occurrence of the events listed in the file.

```
#!/bin/sh

(
echo "Interface name: " $1
echo "Event: " $2

case $2 in
"BOUND")
     echo "Address acquired from server "\
         `/usr/sbin/dhcpinfo -i $1 ServerID`
     ;;
"BOUND6")
```

例 1 Example Event Program *(続き)*

```
     echo "Addresses acquired from server " \
         `/usr/sbin/dhcpinfo -v6 -i $1 ServerID`
     ;;
"EXTEND")
    echo "Lease extended for " \
         `/usr/sbin/dhcpinfo -i $1 LeaseTime`" seconds"
     ;;
"EXTEND6")
    echo "New lease information obtained on $i"
     ;;
"EXPIRE" | "DROP" | "RELEASE")
     ;;

esac
) >/var/run/dhcp_eventhook_output 2>&1
```

Note the redirection of stdout and stderr to a file.

ファイル

`/etc/dhcp/`*`if`*`.dhc`
`/etc/dhcp/`*`if`*`.dh6`
: Contains the configuration for interface. The mere existence of this file does not imply that the configuration is correct, since the lease might have expired. On start-up, `dhcpagent` confirms the validity of the address using REQUEST (for DHCPv4) or Confirm (DHCPv6).

`/etc/dhcp/duid`
`/etc/dhcp/iaid`
: Contains persistent storage for DUID (DHCP Unique Identifier) and IAID (Identity Association Identifier) values. The format of these files is undocumented, and applications should not read from or write to them.

`/etc/default/dhcpagent`
: Contains default values for tunable parameters. All values may be qualified with the interface they apply to by prepending the interface name and a period (“.”) to the interface parameter name. The parameters include: the interface parameter name.

  To configure IPv6 parameters, place the string `.v6` between the interface name (if any) and the parameter name. For example, to set the global IPv6 parameter request list, use `.v6.PARAM_REQUEST_LIST`. To set the `CLIENT_ID` (`DUID`) on `hme0`, use `hme0.v6.CLIENT_ID`.

  The parameters include:

  `VERIFIED_LEASE_ONLY`
  : Indicates that a `RELEASE` rather than a `DROP` should be performed on managed interfaces when the agent terminates. Release causes the client to discard the lease, and the server to make the address available again. Drop causes the client to record the lease in

/etc/dhcp/*interface*.dhc or /etc/dhcp/*interface*.dh6 for later use. In addition, when the link status changes to up or when the system is resumed after a suspend, the client will verify the lease with the server. If the server is unreachable for verification, then the old lease will be discarded (even if it has time remaining) and a new one obtained.

Enabling this option is often desirable on mobile systems, such as laptops, to allow the system to recover quickly from moves.

OFFER_WAIT
: Indicates how long to wait between checking for valid OFFERs after sending a DISCOVER. For DHCPv6, sets the time to wait between checking for valid Advertisements after sending a Solicit.

CLIENT_ID
: Indicates the value that should be used to uniquely identify the client to the server. This value can take one of three basic forms:

decimal,*data*...
0xHHHHH...
"*string*...."

The first form is an RFC 3315 DUID. This is legal for both IPv4 DHCP and DHCPv6. For IPv4, an RFC 4361 Client ID is constructed from this value. In this first form, the format of *data*... depends on the decimal value. The following formats are defined for this first form:

1,*hwtype*,*time*,*lla*
: Type 1, DUID-LLT. The *hwtype* value is an integer in the range 0-65535, and indicates the type of hardware. The *time* value is the number of seconds since midnight, January 1st, 2000 UTC, and can be omitted to use the current system time. The *lla* value is either a colon-separated MAC address or the name of a physical interface. If the name of an interface is used, the *hwtype* value can be omitted. For example: 1,,,hme0

2,*enterprise*,*hex*...
: Type 2, DUID-EN. The *enterprise* value is an integer in the range 0-4294967295 and represents the SMI Enterprise number for an organization. The *hex* string is an even-length sequence of hexadecimal digits.

3,*hwtype*,*lla*
: Type 3, DUID-LL. This is the same as DUID-LLT (type 1), except that a time stamp is not used.

*,*hex*
: Any other type value (0 or 4-65535) can be used with an even-length hexadecimal string.

The second and third forms of CLIENT_ID are legal for IPv4 only. These both represent raw Client ID (without RFC 4361), in hex, or NVT ASCII string format. Thus, "Sun" and 0x53756E are equivalent.

PARAM_REQUEST_LIST
Specifies a list of comma-separated integer values of options for which the client would like values, or symbolic Site or Option option names. Symbolic option names for IPv4 are resolved through /etc/dhcp/inittab. Option names for IPv6 are resolved by means of /etc/dhcp/inittab6.

PARAM_IGNORE_LIST
Specifies a list of options (constructed in the same manner as PARAM_REQUEST_LIST) that the DHCP client will ignore. Ignored options are treated as though the server did not return the options specified. Ignored options are not visible using dhcpinfo(1) or acted on by the client. This parameter can be used, for example, to disable an unwanted client name or default router.

/etc/dhcp/eventhook
Location of a DHCP event program.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

dhcpinfo(1), ifconfig(1M), init(1M), in.mpathd(1M), in.ndpd(1M), syslog(3C), attributes(5), dhcp(5)

『Oracle Solaris の管理: IP サービス』

Croft, B. and Gilmore, J.,Bootstrap Protocol (BOOTP)RFC 951, Network Working Group, September 1985.

Droms, R., Dynamic Host Configuration Protocol, RFC 2131, Network Working Group, March 1997.

Lemon, T. and B. Sommerfeld. RFC 4361, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4). Nominum and Sun Microsystems. February 2006.

Droms, R. RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6). Cisco Systems. July 2003.

注意事項

The dhcpagent daemon can be used on IPv4 logical interfaces, just as with physical interfaces. When used on a logical interface, the daemon automatically constructs a Client ID value based on the DUID and IAID values, according to RFC 4361. The /etc/default/dhcpagent CLIENT_ID value, if any, overrides this automatic identifier.

Unlike physical IPv4 interfaces, dhcpagent does not add or remove default routes associated with logical interfaces.

DHCP can be performed on IPMP IP interfaces to acquire and maintain IPMP data addresses. Because an IPMP IP interface has no hardware address, the daemon automatically constructs a Client ID using the same approach described above for IPv4 logical interfaces. In addition, the lack of a hardware address means the daemon must set the "broadcast" flag in all DISCOVER and REQUEST messages on IPMP IP interfaces. Some DHCP servers may refuse such requests.

DHCP can be performed on IP interfaces that are part of an IPMP group (to acquire and maintain test addresses). The daemon will automatically set the NOFAILOVER and DEPRECATED flags on each test address. Additionally, the daemon will not add or remove default routes in this case. Note that the actual DHCP packet exchange may be performed over any active IP interface in the IPMP group. It is strongly recommended that test addresses have infinite leases. Otherwise, an extended network outage detectable only by probes may cause test address leases to expire, causing in.mpathd(1M) to revert to link-based failure detection and trigger an erroneous repair.

名前

dhcpconfig – DHCP サービス構成ユーティリティー

形式

```
dhcpconfig -D -r resource -p path [-u uninterpreted]
     [-l lease_length] [-n ] [-d DNS_domain]
     [-a DNS_server_addresses] [-h hosts_resource]
     [-y hosts_domain]

dhcpconfig -R server_addresses

dhcpconfig -U [-f] [-x] [-h]

dhcpconfig -N network_address [-m subnet_mask] [-b ]
     [-t router_addresses] [-y NIS-domain]
     [-a NIS_server_addresses] [-g]

dhcpconfig -C -r resource -p path [-f] [-k]
     [-u uninterpreted]

dhcpconfig -X filename [-m macro_list] [-o option_list]
     [-a network_addresses] [-f] [-x] [-g]

dhcpconfig -I filename [-f] [-g]

dhcpconfig -P [parameter[=value],]...

dhcpconfig -S [-f] [-e | -d | -r | -q]
```

機能説明

dhcpconfig コマンドは、動的ホスト構成プロトコル (DHCP) サービスまたは BOOTP リレーサービスを設定および管理するために使用します。経験豊かな Solaris システム管理者を対象としており、スクリプトで簡単に使用できるように設計されています。経験の浅い管理者や、DHCP サービスまたは BOOTP リレーサービスの設定と管理にグラフィカルなユーティリティーを使用したいユーザーには、dhcpmgr ユーティリティーをお勧めします。

dhcpconfig コマンドは、root または DHCP 管理プロファイルに割り当てられたユーザーが実行できます。rbac(5) および user_attr(4) を参照してください。

dhcpconfig には、-D、-R、-U、-N、-C、-X、-I、-P、-S のいずれかの機能フラグを指定する必要があります。

dhcpconfig のメニュー駆動モードは、Solaris 8 以前のバージョンでサポートされています。

dhcpconfig が取得する構成情報の場所

dhcpconfig は、Solaris マシン上のさまざまな構成ファイルを走査して、dhcptab 構成テーブルに追加するマクロに含まれるオプションに値を割り当てるために使用できる情報を取得します。次の表に、dhcpconfig に必要な情報、情報のソース、および情報の使用方法を一覧表示します。

情報	ソース	使用する場所

タイムゾーン	システムの日付とタイムゾーンの設定	Locale マクロ
DNS パラメータ	nsswitch.conf、/etc/resolv.conf	サーバーマクロ
NIS のパラメータ	システムのドメイン名、nsswitch.conf、NIS	ネットワークマクロ
サブネットマスク	ネットワークインタフェース、ネームサービスの netmasks テーブル	ネットワークマクロ

サーバーマシンに前述のパラメータを設定していない場合は、dhcpconfig を使って DHCP サーバーを構成する前に各パラメータを設定してください。dhcpconfig -D でコマンド行にオプションを指定した場合は、指定した値がシステムファイルの値より優先されます。

dhcpconfig ユーティリティーは廃止されており、Oracle Solaris の将来のリリースで削除されます。

オプション

次のオプションがサポートされています。

-C

新しいデータストアに変換します。新しいデータストアに対応する形式で DHCP データテーブルが再作成され、新しいデータストアを使用するように DHCP サーバーが設定されます。

次のサブオプションは必須です。

-p *path_to_data*

SUNWfiles および SUNWbinfiles のパスは、UNIX の絶対パス名である必要があります。dhcp_modules(5) を参照してください。

-r *data_resource*

新しいデータストアリソース。SUNWfiles または SUNWbinfiles のいずれかを指定する必要があります。dhcp_modules(5) を参照してください。

次のサブオプションは省略可能です。

-f

確認を求めるプロンプトを出力しません。-f を指定しないと、変換の開始前に警告と確認のプロンプトが表示されます。

-k

変換の完了後も古い DHCP データテーブルを保持します。変換中に問題が発生した場合は、-k サブオプションを指定していなくても、テーブルは削除されません。

-u *uninterpreted*
dhcpconfig では無視されるが、データストアに渡されて解釈されるデータ。この非公開レイヤーは、RESOURCE_CONFIG キーワードを使ったモジュール固有の構成情報に対応します。解釈されないデータは、dhcpsvc.conf(4) の RESOURCE_CONFIG キーワードの内部に格納されます。-u サブオプションは、SUNWfiles および SUNWbinfiles データストアと同時に指定できません。dhcp_modules(5) を参照してください。

-D
DHCP サービスを構成します。

次のサブオプションは必須です。

-r *data_resource*
SUNWfiles または SUNWbinfiles のいずれかを指定する必要があります。その他のデータストアを使用できる場合があります。dhcp_modules(5) を参照してください。

-p *path*
SUNWfiles および SUNWbinfiles のパスは、UNIX の絶対パス名である必要があります。dhcp_modules(5) を参照してください。

次のサブオプションは省略可能です。

-a *DNS_servers*
DNS サーバーの IP アドレスをコンマで区切って指定します。

-d *DNS_domain*
DNS ドメイン名。

-h *hosts_resource*
ホストデータを格納するリソース。通常は、サーバーで使用されているネームサービスです。有効な値は、files または dns です。

-l *seconds*
リース期間が指定されていないアドレスに適用されるリース期間 (秒単位)。

-n
ネゴシエーション不可能なリース

-y *hosts_domain*
ホストデータとして使用する DNS ドメイン名。h サブオプションに -dns を指定した場合にのみ有効です。

-u *uninterpreted*
dhcpconfig では無視されるが、データストアに渡されて解釈されるデータ。この非公開レイヤーは、RESOURCE_CONFIG キーワードを使ったモジュール固有の構成情報に対応します。解釈されないデータは、dhcpsvc.conf(4) の RESOURCE_CONFIG キーワードの内部に格納されます。-u サブオプションは、SUNWfiles および SUNWbinfiles データストアと同時に指定

できません。dhcp_modules(5) を参照してください。

-I *filename*
事前に Solaris DHCP サーバーからエクスポートしたデータを含む *filename* からデータをインポートします。インポートしたあとで、正しいドメイン名を指定するためにマクロを編集したり、インポートしたネットワークのアドレスの所有サーバーを変更するためにネットワークテーブルを編集したりする必要がある場合があります。そのためには、dhtadm および pntadm を使用します。

次のサブオプションがサポートされています。

-f
重複するデータをインポートされたデータで置き換えます。

-g
インポートの完了後に dhcptab の再読み込みを指示するシグナルをデーモンに送信します。

-N *net_address*
DHCP サービス用の追加ネットワークを構成します。

次のサブオプションがサポートされています。

-a *NIS_server_addresses*
NIS サーバーの IP アドレスのリスト。

-b
ネットワークがポイントツーポイント (PPP) ネットワークであるため、ブロードキャストアドレスを構成しません。-b を使用しない場合、ネットワークは LAN とみなされ、ネットワークアドレスとサブネットマスクを使ってブロードキャストアドレスが決定されます。

-g
dhcptab の再読み込みを指示するシグナルをデーモンに送信します。

-m *xxx.xxx.xxx.xxx*
ネットワークのサブネットマスク。-m を使用しない場合は、ネットマスクからサブネットマスクが取得されます。

-t *router_addresses*
ルーター IP アドレスのリスト。指定しない場合は、ルーター検索フラグが設定されます。

-y *NIS_domain_name*
このネットワークで NIS が使用されている場合に、NIS のドメイン名を指定します。

-P
DHCP サービスのパラメータを構成します。個々のパラメータと値を次のパターンで指定します。

parameter[=*value*],...

パラメータと値の詳細は次のとおりです。

parameter
dhcpsvc.conf(4) に示されているいずれかの DHCP サービスパラメータ。対応する *value* を指定しないと、現在のパラメータ値が表示されます。*parameter* を指定しないと、すべてのパラメータと現在の値が表示されます。

value
サーバーパラメータを設定するための省略可能な文字列 (設定可能な値の場合)。値を指定していないか、値が空 ("") の場合は、そのパラメータと現在の値が削除されます。

パラメータの変更後に新しいパラメータ値を使用するには、DHCP サーバーを再起動する必要があります。

-R *server_addresses*
BOOTP リレーサービスを構成します。BOOTP 要求または DHCP 要求は、このリストで指定した各サーバーに転送されます。

server_addresses は、ホスト名または IP アドレス、あるいはその両方をコンマで区切ったリストです。

-S
DHCP サービスを制御します。

次のサブオプションがサポートされています。

-d
DHCP サービスを無効にして停止します。

-e
DHCP サービスを有効にして起動します。

-q
DHCP サービスの状態を表示します。ステータスは終了ステータスにエンコードされています。

```
0    DHCP service disabled and stopped
1    DHCP service enabled and stopped
2    DHCP service disabled and running
3    DHCP service enabled and running
```

-r
DHCP サービスを有効にして再起動します。

-U
DHCP サービスまたは BOOTP リレーサービスの構成を解除します。

次のサブオプションがサポートされています。

-f
確認を求めるプロンプトを出力しません。-f を指定しないと、警告と確認のプロンプトが表示されます。

-h
ネームサービスからホストのエントリを削除します。

-x
dhcptab とネットワークテーブルを削除します。

-X *filename*
別の Solaris DHCP サーバーにデータを移動するため、DHCP データテーブルのデータをエクスポートして *filename* に保存します。

次のサブオプションは省略可能です。

-a *networks_to_export*
アドレスをエクスポートする必要があるネットワークのリスト、またはすべてのネットワークを示すキーワード ALL。-a を指定しないと、ネットワークはエクスポートされません。

-g
エクスポートの完了後に dhcptab の再読み込みを指示するシグナルをデーモンに送信します。

-m *macros_to_export*
エクスポートするマクロのリスト、またはすべてのマクロを示すキーワード ALL。-m を指定しないと、マクロはエクスポートされません。

-o *options_to_export*
エクスポートするオプションのリスト、またはすべてのオプションを示すキーワード ALL。-o を指定しないと、オプションはエクスポートされません。

-x
エクスポート後、このサーバーからデータを削除します。-x を指定しないと、実質的にデータをコピーすることになります。

使用例

例 1 バイナリファイルのデータストアを使って DHCP サービスを構成する

次のコマンドは、リース期間を 28800 秒 (8 時間) とし、DNS ドメイン acme.eng のバイナリファイルのデータストアを使って DHCP サービスを構成します。

```
example# dhcpconfig -D -r SUNWbinfiles -p /var/dhcp -l 28800\
             -d acme.eng -a 120.30.33.4 -h dns -y acme.eng
```

例 2 BOOTP リレーエージェントを構成する

次のコマンドは、DHCP デーモンを BOOTP リレーエージェントとして構成します。このエージェントは IP アドレスが 120.30.33.7 のサーバーと 120.30.42.132 のサーバーに BOOTP 要求と DHCP 要求を転送します。

例 2 BOOTP リレーエージェントを構成する (続き)

```
example# dhcpconfig -R 120.30.33.7,120.30.42.132
```

例 3 DHCP サービスの構成を解除する

次のコマンドは、確認付きで DHCP サービスの構成を解除し、DHCP データテーブルとホストテーブルのエントリを削除します。

```
example# dhcpconfig -U -x -h
```

例 4 DHCP サービス用のネットワークを構成する

次のコマンドは、DHCP サービス用の追加の LAN ネットワークを構成します。クライアントがルーター検索を使用するように設定し、NIS ドメイン名と NIS サーバーアドレスを指定します。

```
example# dhcpconfig -N 120.30.171.0 -y east.acme.eng.com\
             -a 120.30.33.4
```

例 5 DHCP サーバーからネットワーク、マクロ、およびオプションをエクスポートする

次のコマンドは、1 つのネットワーク (120.30.171.0) とその内部のアドレス、マクロ 120.30.171.0、およびオプション motd と PSptr を DHCP サーバーからエクスポートし、エクスポートしたデータを /export/var/120301710_data というファイルに保存したあと、エクスポートしたデータをサーバーから削除します。

```
example# dhcpconfig -X /var/dhcp/120301710_export
     -a 120.30.171.0 -m 120.30.171.0 -o motd,PSptr
```

例 6 DHCP サーバーにデータをインポートする

次のコマンドは、以前に Solaris DHCP サーバーからエクスポートされたデータを格納する /net/golduck/export/var/120301710_data というファイルから DHCP データをインポートし、インポート先のサーバー上の衝突するデータを上書きし、インポートの完了後に dhcptab の再読み込みを指示するシグナルをデーモンに送信します。

```
example# dhcpconfig -I /net/golduck/export/var/120301710_data -f -g
```

例 7 DHCP サーバーのパラメータを設定する

次のコマンドは、DHCP クライアント上の DNS 情報の更新がタイムアウトするまでの DHCP サーバーの待機時間を 5 分に設定します。

```
example# example# dhcpconfig -P UPDATE_TIMEOUT=5
```

例 8 DHCP サーバーを再起動する

次のコマンドは、DHCP サーバーを停止してから再起動します。

例 8 DHCP サーバーを再起動する (続き)

```
example#  example# dhcpconfig -S -r
DHCP server stopped
DHCP server started
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dhcp
インタフェースの安定性	廃止

関連項目

dhcpmgr(1M), dhtadm(1M), in.dhcpd(1M), pntadm(1M), dhcp_network(4), dhcptab(4), dhcpsvc.conf(4), nsswitch.conf(4), resolv.conf(4), user_attr(4), attributes(5), dhcp(5), dhcp_modules(5), rbac(5)

『Oracle Solaris の管理: IP サービス』

名前

dhcpmgr – graphical interface for managing DHCP service

形式

`/usr/sadm/admin/bin/dhcpmgr`

機能説明

`dhcpmgr` is a graphical user interface which enables you to manage the Dynamic Host Configuration Protocol (DHCP) service on the local system. It performs the functions of the `dhcpconfig`, `dhtadm`, and `pntadm` command line utilities. You must be `root` to use `dhcpmgr`. The `dhcpmgr` Help, available from the `Help` menu, contains detailed information about using the tool.

The `dhcpmgr` utility is obsolete and is subject to removal in a future release of Oracle Solaris.

使用法

You can perform the following tasks using `dhcpmgr`:

Configure DHCP service	Use `dhcpmgr` to configure the DHCP daemon as a DHCP server, and select the data store to use for storing network configuration tables..
Configure BOOTP relay service	Use `dhcpmgr` to configure the DHCP daemon as a BOOTP relay.
Manage DHCP or BOOTP relay service	Use `dhcpmgr` to start, stop, enable, disable or unconfigure the DHCP service or BOOTP relay service, or change DHCP server parameters.
Manage DHCP addresses	Use `dhcpmgr` to add, modify, or delete IP addresses leased by the DHCP service.
Manage DHCP macros	Use `dhcpmgr` to add, modify or delete macros used to supply configuration parameters to DHCP clients.
Manage DHCP options	Use `dhcpmgr` to add, modify or delete options used to define parameters deliverable through DHCP.
Convert to a new DHCP data store	Use `dhcpmgr` to configure the DHCP server to use a different data store, and convert the DHCP data to the format used by the new data store.
Move DHCP data to another server	Use `dhcpmgr` to export data from one Solaris DHCP server and import data onto another Solaris DHCP server.

終了ステータス

The following exit values are returned:

`0`	Successful completion.
`non-zero`	An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/dhcp/dhcpmgr
Interface Stability	Obsolete

関連項目

dhcpconfig(1M), dhtadm(1M), pntadm(1M), in.dhcpd(1M), dhcpsvc.conf(4), dhcp_network(4), dhcptab(4), attributes(5), dhcp(5), dhcp_modules(5)

『Oracle Solaris DHCP Service Developer's Guide』

『Oracle Solaris の管理: IP サービス』

名前

dhtadm – DHCP configuration table management utility

形式

```
dhtadm -C [-r resource] [-p path] [-u uninterpreted] [-g]

dhtadm -A -s symbol_name -d definition [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -A -m macro_name -d definition [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -M -s symbol_name -d definition [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -M -s symbol_name -n new_name [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -M -m macro_name -n new_name [-r resource] [-p path]
     [-u uninterpreted] [-g]

dhtadm -M -m macro_name -d definition [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -M -m macro_name -e symbol=value [-r resource]
     [-p path] [-u uninterpreted] [-g]

dhtadm -D -s symbol_name [-r resource] [-p path]
     [-u uninterpreted] [-g]

dhtadm -D -m macro_name [-r resource] [-p path]
     [-u uninterpreted] [-g]

dhtadm -P [-r resource] [-p path] [-u uninterpreted] [-g]

dhtadm -R [-r resource] [-p path] [-u uninterpreted] [-g]

dhtadm -B [-v] [batchfile] [-g]
```

機能説明

`dhtadm` manages the Dynamic Host Configuration Protocol (DHCP) service configuration table, `dhcptab`. You can use it to add, delete, or modify DHCP configuration macros or options or view the table. For a description of the table format, see `dhcptab(4)`.)

The `dhtadm` command can be run by root, or by other users assigned to the DHCP Management profile. See `rbac(5)` and `user_attr(4)`.

After you make changes with `dhtadm`, you should issue a `SIGHUP` to the DHCP server, causing it to read the `dhcptab` and pick up the changes. Do this using the `-g` option.

The `dhtadm` utility is obsolete and is subject to removal in a future release of Oracle Solaris.

オプション

One of the following function flags must be specified with the `dhtadm` command: `-A`, `-B`, `-C`, `-D`, `-M`, `-P` or `-R`.

The following options are supported:

`-A`

Add a symbol or macro definition to the `dhcptab` table.

The following sub-options are required:

`-d` *definition*

Specify a macro or symbol definition.

definition must be enclosed in single quotation marks. For macros, use the form `-d ':`*symbol*`=`*value*`:`*symbol*`=`*value*`:'`. Enclose a *value* that contains colons in double quotation marks. For symbols, the definition is a series of fields that define a symbol's characteristics. The fields are separated by commas. Use the form `-d '`*context*`,`*code*`,`*type*`,`*granularity*`,`*maximum*`'`. See `dhcptab(4)` for information about these fields.

`-m` *macro_name*

Specify the name of the macro to be added.

The `-d` option must be used with the `-m` option. The `-s` option cannot be used with the `-m` option.

`-s` *symbol_name*

Specify the name of the symbol to be added.

The `-d` option must be used with the `-s` option. The `-m` option cannot be used with the `-s` option.

`-B`

Batch process `dhtadm` commands. `dhtadm` reads from the specified file or from standard input a series of `dhtadm` commands and execute them within the same process. Processing many `dhtadm` commands using this method is much faster than running an executable batchfile itself. Batch mode is recommended for using `dhtadm` in scripts.

The following sub-option is optional:

`-v`

Display commands to standard output as they are processed.

`-C`

Create the DHCP service configuration table, `dhcptab`.

`-D`

Delete a symbol or macro definition.

The following sub-options are required:

`-m` *macro_name*

Delete the specified macro.

`-s` *symbol_name*

Delete the specified symbol.

`-g`
Signal the DHCP daemon to reload the `dhcptab` after successful completion of the operation.

`-M`
Modify an existing symbol or macro definition.

The following sub-options are required:

`-d` *definition*
Specify a macro or symbol definition to modify.

The definition must be enclosed in single quotation marks. For macros, use the form `-d ':`*symbol*`=`*value*`:`*symbol*`=`*value*`:'`. Enclose a *value* that contains colons in double quotation marks. For symbols, the definition is a series of fields that define a symbol's characteristics. The fields are separated by commas. Use the form `-d '`*context*`,`*code*`,`*type*`,`*granularity*`,`*maximum*`'`. See `dhcptab(4)` for information about these fields.

`-e`
This sub-option uses the *symbol* =*value* argument. Use it to edit a *symbol*/*value* pair within a macro. To add a symbol which does not have an associate value, enter:

symbol=`_NULL_VALUE_`

To delete a symbol definition from a macro, enter:

symbol=

`-m`
This sub-option uses the *macro_name* argument. The `-n`, `-d`, or `-e` sub-options are legal companions for this sub-option..

`-n`
This sub-option uses the *new_name* argument and modifies the name of the object specified by the `-m` or `-s` sub-option. It is not limited to macros. . Use it to specify a new macro name or symbol name.

`-s`
This sub-option uses the *symbol_name* argument. Use it to specify a symbol. The `-d` sub-option is a legal companion.

`-p` *path*
Override the `dhcpsvc.conf(4)` configuration value for `PATH=` with *path*. See `dhcpsvc.conf(4)` for more details regarding *path*. See `dhcp_modules(5)` for information regarding data storage modules for the DHCP service.

`-P`
Print (display) the `dhcptab` table.

-r *data_store_resource*
Override the dhcpsvc.conf(4) configuration value for RESOURCE= with the *data_store_resource* specified. See dhcpsvc.conf(4) for more details on resource type. See 『Oracle Solaris DHCP Service Developer's Guide』 for more information about adding support for other data stores. See dhcp_modules(5) for information regarding data storage modules for the DHCP service.

-R
Remove the dhcptab table.

-u *uninterpreted*
Data which is ignored by dhtadm, but passed to currently configured public module, to be interpreted by the data store. The private layer provides for module-specific configuration information through the use of the RESOURCE_CONFIG keyword. Uninterpreted data is stored within RESOURCE_CONFIG keyword of dhcpsvc.conf(4). See dhcp_modules(5) for information regarding data storage modules for the DHCP service.

使用例

例 1 Creating the DHCP Service Configuration Table

The following command creates the DHCP service configuration table, dhcptab:

```
# dhtadm -C
```

例 2 Adding a Symbol Definition

The following command adds a Vendor option symbol definition for a new symbol called MySym to the dhcptab table in the SUNWfiles resource in the /var/mydhcp directory:

```
# dhtadm -A -s MySym
   -d 'Vendor=SUNW.PCW.LAN,20,IP,1,0'
   -r SUNWfiles -p /var/mydhcp
```

例 3 Adding a Macro Definition

The following command adds the aruba macro definition to the dhcptab table. Note that symbol/value pairs are bracketed with colons (:).

```
# dhtadm -A -m aruba \
     -d ':Timeserv=10.0.0.10 10.0.0.11:DNSserv=10.0.0.1:'
```

例 4 Modifying a Macro Definition

The following command modifies the Locale macro definition, setting the value of the UTCOffst symbol to 18000 seconds. Note that any macro definition which includes the definition of the Locale macro inherits this change.

```
# dhtadm -M -m Locale -e 'UTCOffst=18000'
```

例 5 Deleting a Symbol

The following command deletes the `Timeserv` symbol from the `aruba` macro. Any macro definition which includes the definition of the `aruba` macro inherits this change.

```
# dhtadm -M -m aruba -e 'Timeserv='
```

例 6 Adding a Symbol to a Macro

The following command adds the `Hostname` symbol to the `aruba` macro. Note that the `Hostname` symbol takes no value, and thus requires the special value `_NULL_VALUE_`. Note also that any macro definition which includes the definition of the `aruba` macro inherits this change.

```
# dhtadm -M -m aruba -e 'Hostname=_NULL_VALUE_'
```

例 7 Renaming a Macro

The following command renames the `Locale` macro to `MyLocale`. Note that any `Include` statements in macro definitions which include the `Locale` macro also need to be changed.

```
# dhtadm -M -m Locale -n MyLocale
```

例 8 Deleting a Symbol Definition

The following command deletes the `MySym` symbol definition. Note that any macro definitions which use `MySym` needs to be modified.

```
# dhtadm -D -s MySym
```

例 9 Printing a dhcptab

The following command prints to standard output the contents of the `dhcptab` that is located in the data store and path indicated in the `dhcpsvc.conf` file:.

```
# dhtadm -P
```

例 10 Executing dhtadm in Batch Mode

The following command runs a series of `dhtadm` commands contained in a batch file and signals the daemon to reload the `dhcptab` once the commands have been executed: :

```
# dhtadm -B addmacros -g
```

終了ステータス

`0`
Successful completion.

`1`
Object already exists.

`2`
Object does not exist.

3
Non-critical error.

4
Critical error.

ファイル

`/etc/inet/dhcpsvc.conf`
contains service configuration parameters for the DHCP service

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dhcp
Interface Stability	Obsolete

関連項目

`dhcpconfig(1M)`, `dhcpmgr(1M)`, `in.dhcpd(1M)`, `dhcpsvc.conf(4)`, `dhcp_network(4)`, `dhcptab(4)`, `hosts(4)`, `user_attr(4)`, `attributes(5)`, `dhcp(5)`, `dhcp_modules(5)rbac(5)`

『Oracle Solaris DHCP Service Developer's Guide』

『Oracle Solaris の管理: IP サービス』

Alexander, S., and R. Droms, DHCP Options and BOOTP Vendor Extensions, RFC 1533, Lachman Technology, Inc., Bucknell University, October 1993.

Droms, R., Interoperation Between DHCP and BOOTP, RFC 1534, Bucknell University, October 1993.

Droms, R., Dynamic Host Configuration Protocol, RFC 1541, Bucknell University, October 1993.

Wimer, W., Clarifications and Extensions for the Bootstrap Protocol, RFC 1542, Carnegie Mellon University, October 1993.

名前

dig – DNS 検索ユーティリティー

形式

```
dig [@server] [-b address] [-c class] [-f filename]
    [-k filename] [-m] [-p port#] [-q name] [-t type] [-x addr]
    [-y [hmac:]name:key] [-4] [-6] [name] [type] [class] [queryopt]...

dig [-h]

dig [global-queryopt...] [query...]
```

機能説明

dig (Domain Information Groper) ユーティリティーは、DNS ネームサーバーに問い合わせるための柔軟性に優れたツールです。dig ユーティリティーは、DNS 検索を実行し、照会したネームサーバーから返された回答を表示します。dig ユーティリティーは、柔軟性に優れ、使いやすく、出力が明確であるため、ほとんどの DNS 管理者がこのユーティリティーを使用して DNS の問題をトラブルシューティングします。ほかの検索ツールは、多くの場合で dig ほど機能性に優れていません。

通常、dig は、コマンド行引数とともに使用されますが、ファイルから検索要求を読み取るバッチモード操作にも対応します。-h オプションを指定すると、そのコマンド行引数とオプションの簡単な要約が出力されます。以前のバージョンとは異なり、BIND 9 における dig の実装では、コマンド行から複数の検索を発行できます。

特定のネームサーバーに照会するよう指示されていないかぎり、dig は、/etc/resolv.conf に記載されている各サーバーに照会します。

コマンド行引数またはオプションを指定しない場合、dig は、「.」(ルート) に対して NS クエリーを実行します。

${HOME}/.digrc を使用すると、dig のユーザーごとのデフォルトを設定できます。コマンド行引数の前に、このファイルが読み取られ、含まれているオプションが適用されます。

クラス名 IN および CH は、トップレベルドメイン名 IN および CH と一致します。これらのトップレベルドメインを検索するときは、-t オプションおよび -c オプションを使用してタイプとクラスを指定するか、"IN." および "CH." を使用します。

簡単な使用法

一般的な dig の呼び出しを次に示します。

```
dig @server name type
```

各表記の意味は次のとおりです。

server

照会するネームサーバーの名前または IP アドレス。ドット区切り 10 進表記の IPv4 アドレスか、コロン区切り表記の IPv6 アドレスを指定できます。指定された *server* 引数がホスト名の場合、dig は、その名前を解決してから、そのネームサーバーに照会します。*server* 引数を指定しない場合、dig は /etc/resolv.conf を参照し、そのファイルに記載されているネームサーバーに照会します。応答したネームサーバーからの応答が表示されます。

name
検索するリソースレコードの名前。

type
どのタイプのクエリーが必要かを示します (ANY、A、MX、SIG など)。*type* には、任意の有効なクエリータイプを指定できます。*type* 引数を指定しない場合、`dig` は、A レコードの検索を実行します。

オプション

次のオプションがサポートされています。

`-4`
IPv4 トランスポートのみを使用します。デフォルトでは、IPv4 トランスポートと IPv6 トランスポートの両方を使用できます。オプション `-4` とオプション `-6` は相互に排他的です。

`-6`
IPv6 トランスポートのみを使用します。デフォルトでは、IPv4 トランスポートと IPv6 トランスポートの両方を使用できます。オプション `-4` とオプション `-6` は相互に排他的です。

`-b` *address*
クエリーのソース IP アドレスを *address* に設定します。これは、ホストのいずれかのネットワークインタフェース上の有効なアドレスか、`0.0.0.0` または `::` である必要があります。`#`*<port>* を追加して、オプションのポートを指定できます。

`-c` *class*
デフォルトのクエリークラス (インターネットを表す IN) を上書きします。*class* 引数は、HS (Hesiod レコード) や CH (CHAOSNET レコード) などの有効な任意のクラスです。

`-f` *filename*
ファイル *filename* から、処理対象の検索要求のリストを読み取って、バッチモードで動作します。ファイルには、1 行に 1 つずつ、複数のクエリーが含まれています。ファイル内の各エントリは、コマンド行インタフェースを使用して `dig` で発行するクエリーを指定する場合と同じように構成するようにしてください。

`-h`
コマンド行引数およびオプションの簡単な要約を出力します。

`-k` *filename*
`dig` によって送信される DNS クエリーとその応答に、トランザクション署名 (Transaction Signature、TSIG) を使用して署名するための TSIG 鍵ファイルを指定します。

`-m`
メモリー使用量のデバッグを有効にします。

-p *port#*
標準以外のポート番号を照会します。*port#* 引数は、dig が標準の DNS ポート番号 53 の代わりにクエリーを送信するポート番号です。このオプションを使用すると、標準以外のポート番号でクエリーを待機するように構成されているネームサーバーをテストできます。

-q *name*
クエリー名を *name* に設定します。これは、クエリー名をほかの引数と容易に区別できるという点で便利です。

-t *type*
クエリータイプを *type* に設定します。これには、BIND9 でサポートされている有効なクエリータイプを指定できます。逆検索を示す -x オプションが指定されていないかぎり、デフォルトのクエリータイプは「A」です。タイプ AXFR を指定することによって、ゾーン転送を要求できます。増分ゾーン転送 (Incremental Zone Transfer、IXFR) が必要な場合は、*type* を ixfr=*N* に設定します。増分ゾーン転送には、ゾーンの SOA レコードのシリアル番号が *N* であった時点以降にゾーンに加えられた変更が含まれます。

-x *addr*
逆検索を簡略化します (アドレスを名前にマッピングします)。*addr* 引数は、ドット区切り 10 進表記の IPv4 アドレスか、コロン区切りの IPv6 アドレスです。このオプションを使用する場合は、*name*、*class*、および *type* の各引数を指定する必要はありません。dig ユーティリティーは、自動的に 11.12.13.10.in-addr.arpa などの名前の検索を実行し、クエリーのタイプとクラスをそれぞれ PTR と IN に設定します。デフォルトでは、IPv6 アドレスはニブル形式を使用して IP6.ARPA ドメインで検索されます。IP6.INT ドメインを使用する以前の RFC1886 方式を使用するには、-i オプションを指定します。ビット文字列ラベル (RFC 2874) は、実験的な段階であり、試行されません。

-y [*hmac*:]*name*:*key*
コマンド行でトランザクション署名 (TSIG) 鍵を指定します。これは、dig によって送信される DNS クエリーと、その応答に署名するために行います。-y オプションを使用すると、TSIG 鍵自体をコマンド行で指定することもできます。オプションの *hmac* は TSIG のタイプであり、デフォルトは HMAC-MD5 です。*name* 引数は TSIG 鍵の名前であり、*key* 引数は実際の鍵です。鍵は base-64 でエンコードされた文字列であり、通常は、dnssec-keygen(1M) によって生成されます。

鍵が ps(1) の出力またはシェルの履歴ファイルに示されるため、複数ユーザーシステムで -y オプションを使用するときは注意するようにしてください。dig で TSIG 認証を使用する場合は、照会されるネームサーバーが、使用される鍵とアルゴリズムを認識できる必要があります。BIND では、named.conf で適切な key 文と server 文を指定します。

クエリーオプション

dig ユーティリティーには、検索方法と結果の表示方法に影響を与える多数のクエリーオプションが用意されています。これらのクエリーオプションを使用すると、クエリーヘッダーのフラグビットを設定/リセットしたり、回答のどの部分を出力するかを決定したり、タイムアウト/再試行方針を決定したりできます。

各クエリーオプションは、プラス記号 (+) とそれに続くキーワードによって識別されます。一部のキーワードは、オプションを設定またはリセットします。これらのキーワードの前に文字列 no を指定すると、そのキーワードの意味を否定できます。タイムアウトの間隔など、オプションに値を割り当てるキーワードもあります。これらは、+keyword= *value* の形式をとります。クエリーオプションを次に示します。

+[no]tcp
: ネームサーバーに照会するときに TCP を使用します [使用しません]。デフォルトの動作では UDP が使用されます。ただし、TCP 接続が使用される AXFR または IXFR クエリーが要求された場合を除きます。

+[no]vc
: ネームサーバーに照会するときに TCP を使用します [使用しません]。+[no]tcp の代替構文は下位互換性を確保するために用意されています。「vc」は、「virtual circuit (仮想回路)」を表します。

+[no]ignore
: TCP で再試行する代わりに、UDP 応答での切り捨てを無視します。デフォルトでは、TCP での再試行が実行されます。

+domain=*somename*
: /etc/resolv.conf の domain 指令で指定された場合と同じように単一のドメイン *somename* を含めるように検索リストを設定し、+search オプションが指定された場合と同じように検索リストが処理されるようにします。

+[no]search
: searchlist、または存在する場合は resolv.conf の domain 指令で定義されている検索リストを使用します [使用しません]。デフォルトでは、検索リストは使用されません。

+[no]showsearch
: 中間結果を表示して検索を実行します [実行しません]。

+[no]defname
: 非推奨、+[no]search の同義語として処理されます。

+[no]aaonly
: クエリーで aa フラグを設定します。

+[no]aaflag
: +[no]aaonly の同義語です。

`+[no]adflag`
クエリーで AD (Authentic Data、認証済みデータ) ビットを設定します [設定しません]。これは、サーバーのセキュリティーポリシーに従って、すべての回答セクションおよび権限セクションが検証されたかどうかにかかわらず、安全であると応答するようにサーバーに要求します。設定 `AD=1` は、すべてのレコードが安全であると検証されいて、回答が `OPT-OUT` の範囲からではないことを示します。`AD=0` は、回答の一部が安全ではないか、検証されていないことを示します。

`+[no]cdflag`
クエリーで CD (Checking Disabled、チェック無効) ビットを設定します [設定しません]。これは、応答の DNSSEC 検証を実行しないようにサーバーに要求します。

`+[no]cl`
レコードを出力するときに CLASS を表示します [表示しません]。

`+[no]ttlid`
レコードを出力するときに TTL を表示します [表示しません]。

`+[no]recurse`
クエリーで RD (Recursion Desired、再帰要望) ビットの設定を切り替えます。デフォルトでは、このビットは設定されています。これは、`dig` で通常は再帰クエリーが送信されることを意味します。`+nssearch` または `+trace` クエリーオプションが使用されている場合、再帰は自動的に無効になります。

`+[no]nssearch`
このオプションを設定すると、`dig` は、検索対象の名前が含まれるゾーンの信頼できるネームサーバーを見つけて、各ネームサーバーが持つゾーンの SOA レコードを表示しようとします。

`+[no]trace`
検索対象の名前の、ルートネームサーバーからの委譲パスのトレースの有効/無効を切り替えます。デフォルトでは、トレースは無効になっています。トレースを有効にすると、`dig` は、反復クエリーを実行して検索対象の名前を解決します。ルートサーバーからのリフェラルを追跡し、検索の解決に使用した各サーバーからの回答を表示します。

`+[no]cmd`
出力の中にある、`dig` のバージョンと適用されているクエリーオプションを識別する先頭のコメントの有無を切り替えます。デフォルトでは、このコメントは出力されます。

`+[no]short`
簡易形式の回答を表示します。デフォルトでは、詳細形式の回答が出力されます。

`+[no]identify`
: *+short* オプションが有効になっている場合に、回答を返したサーバーの IP アドレスおよびポート番号を表示します [表示しません]。短い形式の回答を要求した場合、デフォルトでは、回答を返したサーバーの発信元アドレスおよびポート番号は表示されません。

`+[no]comments`
: 出力でのコメント行の表示の有無を切り替えます。デフォルトでは、コメントが出力されます。

`+[no]stats`
: クエリーが発行された時刻、応答のサイズなどの統計情報の出力の有無を切り替えます。デフォルトの動作では、クエリーの統計情報が出力されます。

`+[no]qr`
: クエリーが送信されたときにクエリーを出力します [出力しません]。デフォルトでは、クエリーは出力されません。

`+[no]question`
: 回答が返されたときに、クエリーの質問セクションを出力します [出力しません]。デフォルトでは、質問セクションがコメントとして出力されます。

`+[no]answer`
: 応答の回答セクションを表示します [表示しません]。デフォルトでは表示されます。

`+[no]authority`
: 応答の権限セクションを表示します [表示しません]。デフォルトでは表示されます。

`+[no]additional`
: 応答の追加セクションを表示します [表示しません]。デフォルトでは表示されます。

`+[no]all`
: すべての表示フラグを設定またはクリアします。

`+time=`*T*
: クエリーのタイムアウトを *T* 秒に設定します。デフォルトのタイムアウトは 5 秒です。*T* を 1 未満に設定しようとした場合、1 秒が適用されます。

`+tries=`*T*
: UDP での最大試行回数を *T* に設定します。デフォルトの数値は 3 (最初の試行が 1 回と、その後の再試行が 2 回) です。T が 0 以下の場合、再試行回数は自動的に 1 に切り上げられます。

`+retry=`*T*
: UDP での再試行回数を *T* に設定します。デフォルトは 2 です。

+ndots=*D*

name に示すドットの数を *D* に設定します。この数のドットが含まれる名前は絶対名と見なされます。デフォルト値は、/etc/resolv.conf で ndots 文を使用して定義されている値です。ndots 文が存在しない場合は 1 です。ドットがこの値よりも少ない名前は、相対名と解釈され、/etc/resolv.conf の search 指令または domain 指令に示されているドメイン内で検索されます。

+bufsize=*B*

EDNS0 を使用して通知される UDP メッセージバッファーサイズを *B* バイトに設定します。このバッファーの最大および最小サイズはそれぞれ 65535 と 0 です。この範囲外の値は、適切に切り上げられるか、切り下げられます。

+edns=#

クエリーで使用される EDNS バージョンを指定します。有効な値の範囲は 0 - 255 です。EDNS バージョンを設定すると、EDNS クエリーが送信されます。+noedns により、記憶されている EDNS バージョンはクリアされます。

+[no]multiline

SOA レコードなどのレコードを、詳細な複数行形式で、人間が読める形式のコメントとともに出力します。デフォルトでは、dig 出力をマシンで容易に解析できるようにするため、各レコードが 1 行に 1 つずつ出力されます。

+[no]fail

SERVFAIL を受け取った場合に、次のサーバーを試行しません。デフォルトでは次のサーバーを試行しません。これは、通常のスタブリゾルバとは逆の動作です。

+[no]besteffort

不正な形式のメッセージの内容を表示しようとします。デフォルトでは、不正な形式の回答は表示されません。

+[no]dnssec

クエリーの追加セクションの OPT レコードで DNSSEC OK (DO) ビットを設定し、DNSSEC レコードが送信されるよう要求します。

+[no]sigchase

DNSSEC 署名チェーンを追跡します。-DDIG_SIGCHASE を使用して dig をコンパイルする必要があります。

+trusted-key=####

+sigchase で使用される信頼できる鍵が含まれるファイルを指定します。DNSKEY レコードは、1 行に 1 つずつ記述する必要があります。

ファイルを指定しない場合、dig は、/etc/trusted-key.key、現在のディレクトリの trusted-key.key の順に検索します。

-DDIG_SIGCHASE を使用して dig をコンパイルする必要があります。

`+[no]topdown`
DNSSEC 署名チェーンを追跡するときに、トップダウン検証を実行します。`-DDIG_SIGCHASE` を使用して `dig` をコンパイルする必要があります。

`+[no]nsid`
クエリーを送信するときに、EDNS ネームサーバーの ID 要求を含めます。

複数のクエリー

BIND 9 における `dig` の実装では、(`-f` バッチファイルオプションのサポートに加えて) コマンド行での複数のクエリーの指定がサポートされます。これらの各クエリーは、それぞれ一連のフラグ、オプション、およびクエリーオプションを設定して発行できます。

この場合、各 *query* 引数は、前述のコマンド行構文での個々のクエリーを表します。各クエリーは、標準のオプションとフラグ、検索対象の名前、クエリータイプ (省略可能)、クラス、およびそのクエリーに適用するクエリーオプションで構成されます。

すべてのクエリーに適用するクエリーオプションのグローバルセットを設定することもできます。これらのグローバルなクエリーオプションは、コマンド行で、名前、クラス、タイプ、オプション、フラグ、およびクエリーオプションの最初の組の前に指定する必要があります。グローバルなクエリーオプション (`+[no]cmd` オプションを除く) は、クエリーごとの一連のクエリーオプションで上書きできます。例:

```
dig +qr www.isc.org any -x 127.0.0.1 isc.org ns +noqr
```

この例は、コマンド行から `dig` を使用して、3 つの検索 (`www.isc.org` の ANY クエリー、127.0.0.1 の逆検索、および `isc.org` の NS レコードのクエリー) を実行する方法を示しています。グローバルなクエリーオプション `+qr` が適用されているため、`dig` によって、発行された最初のクエリーが各検索で表示されます。最後のクエリーには、ローカルなクエリーオプション `+noqr` が指定されています。これは、`dig` によって、`isc.org` の NS レコードを検索したときに、最初のクエリーが出力されないことを意味しています。

ファイル

`/etc/resolv.conf`
リゾルバ構成ファイル

`${HOME}/.digrc`
ユーザー定義構成ファイル

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	network/dns/bind
インタフェースの安定性	「流動的」

関連項目

dnssec-keygen(1M), host(1M), named(1M), nslookup(1M), attributes(5)

RFC 1035

『BIND 9 Administrator's Reference Manual』を参照してください。このマニュアルページの発行日付時点で、このドキュメントは https://www.isc.org/software/bind/documentation から利用できます。

使用上の留意点

クエリーオプションが多すぎる可能性があります。

注意事項

nslookup(1M) および dig で、「Not Implemented」が NOTIMPL ではなく NOTIMP と報告されるようになりました。これは、NOTIMPL を検索するスクリプトに影響します。

名前

directoryserver – front end for the Directory Server (DS)

形式

```
/usr/sbin/directoryserver
     { setup [-f configuration_file] | uninstall}

/usr/sbin/directoryserver
     {start-admin | stop-admin | restart-admin | startconsole}

/usr/sbin/directoryserver [{-s | -server} server-instance ]
     {start | stop | restart}

/usr/sbin/directoryserver { -s |-server } server-instance
     { monitor | saveconfig | restoreconfig | db2index-task |
       ldif2db-task | ldif2db | ldif2ldap | vlvindex | db2ldif |
       db2ldif-task | db2bak | db2bak-task | bak2db | bak2db-task |
       suffix2instance | account-status | account-activate |
       account-inactivate }
     {...}

/usr/sbin/directoryserver nativetoascii | admin_ip | ldif |
     pwdhash | idsktune | mmldif | keyupg
     {...}

/usr/sbin/directoryserver { magt | sagt } {...}

/usr/sbin/directoryserver help [subcommand]
```

機能説明

The `directoryserver` command is a comprehensive, front end to the utility programs provided by the Solaris Directory Server (DS).

Options for the `directoryserver` command itself must appear before the subcommand. Arguments for a subcommand must appear after the subcommand. Subcommands have specific arguments. See `SUBCOMMANDS`.

Subcommands

The following subcommands are supported:

account-inactivate *args*

Inactivates and locks an entry or group of entries.

The `account-inactivate` subcommand supports the following arguments:

[`-D` *rootdn*]
: Directory Server *userDN* with root permissions, such as Directory Manager.

[`-h` *host*]
: Host name of Directory Server. The default value is the full hostname of the machine where Directory Server is installed.

`-I` *DN*
: Entry DN or role DN to activate.

-j *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
[-p *port*]	Directory Server port. The default value is the LDAP port of Directory Server specified at installation time.
-w *password*	Password associated with the user DN. Supplying the password on the command line is visible using the `/bin/ps` command. This is considered insecure. Use with extreme caution. The value - can be used in place the password. The program prompts the user for a password to be entered from the terminal.

account-activate *args*	Activates an entry or group of entries. The `account-activate` subcommand supports the following arguments

-D *rootdn*	Directory Server userDN with root permissions, such as Directory Manager.
-h *host*	Host name of Directory Server. The default value is the full hostname of the machine where Directory Server is installed.
-I DN	Entry DN or role DN to activate.
-j *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.

`-p` *port*	Directory Server port. The default value is the LDAP port of Directory Server specified at installation time.
`-w` *password*	Password associated with the user DN. Supplying the password on the command line is visible using the `/bin/ps` command. This is considered insecure. Use with extreme caution. The value `-` can be used in place the password. The program prompts the user for a password to be entered from the terminal.

account-status *args*

Provides account status information to establish whether an entry or group of entries is inactivated or not.

The `account-status` subcommand supports the following arguments:

`-D` *rootdn*

`-h` *host*	Host name of Directory Server. The default value is the full hostname of the machine where Directory Server is installed.
`-I` *DN*	Entry DN or role DN whose status is required.
`-j` *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
`-p` *port*	Directory Server port. The default value is the LDAP port of Directory Server specified at installation time.
`-w` *password*	Password associated with the rootDN. Supplying the password on the command line is visible using the

	`/bin/ps` command. This is considered insecure. Use with extreme caution. The value `-` can be used in place of the password. The program prompts the user for a password to be entered from the terminal.
admin_ip *args*	Change the IP address of the administrative server in the configuration. The `admin_ip` subcommand supports the following arguments: *dir_mgr_DN* — Directory Manager's DN. *dir_mgr_password* — Directory Manager's password. *old_ip* — Old IP. *new_ip* — New IP. *port_#* — Port number.
bak2db *backup_directory*	Restore the database from the most recent archived backup. Specify *backup_directory* as the backup directory.
bak2db-task *args*	Restore the data to the database. The bak2db-task subcommand supports the following arguments: [`-a` *directory*] — Directory where the backup files are stored. By default it is under `/var/ds5/slapd-serverID/bak` `-D` *rootDN* — User DN with root permissions, such as Directory Manager. The default is the DN of the directory manager which is read from the `nsslapd-root` attribute under `cn=config`.

	-j *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
	[-t *database_type*]	Database type. The only possible database type is `ldbm`.
	[-v]	Verbose mode.
	-w *password*	Password associated with the user DN. Supplying the password on the command line is visible using the `/bin/ps` command. This is considered insecure. Use with extreme caution. The value - can be used in place the password. The program prompts the user for a password to be entered from the terminal.
db2bak-task *args*	Back up the contents of the database. It creates an entry in the directory that launches this dynamic task. An entry is generated based upon the values provided for each option. The `db2bak-task` subcommand supports the following arguments:	
	[-a *directory*]	Directory where the backup files are stored. By default it is under `/var/ds5/slapd-serverID/bak`. The backup file is named according to the year-month-day-hour format (*YYYY_MM_DD_hhmmss*).
	-D *rootDN*	User DN with root permissions, such as Directory

		Manager. The default is the DN of the directory manager which is read from the `nsslapd-root` attribute under `cn=config`.
	`-j` *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
	`-t` *database_type*	Database type. The only possible database type is `ldbm`.
	[`-v`]	Verbose mode.
	`-w` *password*	Password associated with the user DN. Supplying the password on the command line is visible using the `/bin/ps` command. This is considered insecure. Use with extreme caution. The value `-` can be used in place the password. The program prompts the user for a password to be entered from the terminal.
db2bak [*backup_directory*]	Create a backup of the current database contents. The server must be stopped to run this subcommand. The default is `/var/ds5/slapd-serverID/bak`. The backup file is named according to the year-month-day-hour format (*YYYY_MM_DD_hhmmss*).	
db2index-text *args*	Create and generate the new set of indexes to be maintained following the modification of indexing entries in the `cn=config` configuration file. The `db2index-text` subcommand supports the following arguments:	

`-D` *rootdn*	User DN with root permissions, such as Directory Manager.
`-j` *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
`-n` *backend_instance*	Instance to be indexed.
[`-t` *attributeName*]	Name of the attribute to be indexed. If omitted, all indexes defined for that instance are generated.
[`-v`]	Verbose mode.
`-w` *password*	Password associated with the user DN. Supplying the password on the command line is visible using the `/bin/ps` command. This is considered insecure. Use with extreme caution. The value `-` can be used in place the password. The program prompts the user for a password to be entered from the terminal.

db2ldif-task *args*

Exports the contents of the database to LDIF. It creates an entry in the directory that launches this dynamic task. The entry is generated based upon the values you provide for each option. To run this subcommand the server must be running and either `-n` *backend_instance* or `-s` *include* suffix is required.

The `db2ldif-task` subcommand supports the following arguments:

[`-a` *outputfile*]	File name of the output LDIF file.

-C	Only the main db file is used.
-D *rootDN*	User DN with root permissions, such as Directory Manager.
-j *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure.Use with extreme caution.
[-M]	Output LDIF is stored in multiple files.
[-m]	Minimal base 64 encoding.
{-n *backend_instance*}*	Instance to be exported.
[-N]	Minimal base 64 encoding.
[-o]	Output LDIF to be stored in one file by default with each instance stored in *instance_file* name.
[-r]	Export replica.
[-s]*includesuffix*}*	Suffix(es) to be included or to specify the subtrees to be included if -n has been used.
[-u]	Request that the unique ID is not exported.
[-U]	Request that the output LDIF is not folded.
-w *password*	Password associated with the user DN. Supplying the password on the command line is visible

using the `/bin/ps` command. This is considered insecure. Use with extreme caution.

The value `-` can be used in place the password. The program prompts the user for a password to be entered from the terminal.

{`-x` *excludesuffix*}*

Suffixes to be excluded.

[`-1`]

Delete, for reasons of backward compatibility the first line of the LDIF file that gives the version of the LDIF standard.

db2ldif *args*

Export the contents of the database to LDIF. You must specify either the `-n` or the `-s` option or both.

The `db2ldif` subcommand supports the following options:

[`-a` *outputfile*]

File name of the output LDIF file.

[`-C`]

Only use the main db file.

[`-m`]

Minimal base64 encoding.

[`-M`]

Use of several files for storing the output LDIF with each instance stored in *instance_file name* (where file name is the file name specified for `-a` option).

{`-n` *baclemd_instance*}*

Instance to be exported.

[`-N`]

Specify that the entry IDs are not to be included in the LDIF output. The entry IDs are necessary

	only if the `db2ldif` output is to be used as input to `db2index-text`.
[`-r`]	Export replica.
{`-s` *includesuffix*}*	Suffixes to be included or to specify the subtrees to be included if `-n` has been used.
[{`-x` *excludesuffix*}]*	Suffixes to be excluded.
[`-u`]	Request that the unique id is not exported.
[`-U`]	Request that the output LDIF is not folded.
[`-1`]	Delete, for reasons of backward compatibility, the first line of the LDIF file which gives the version of the LDIF standard.

help [*subcommand*]

Display `directoryserver` usage message or subcommand specific usage message.

idsktune *args*

Provide an easy and reliable way of checking the patch levels and kernel parameter settings for your system. You must install the Directory Server before you can run `idsktune`. It gathers information about the operating system, kernel, and TCP stack to make tuning recommendations.

The `idsktune` subcommand supports the following arguments:

[`-c`]	Client-specific tuning: the output only includes tuning recommendations for running a directory client application.
[`-D`]	Debug mode: the output includes the commands it runs internally, preceded by `DEBUG` heading.
[`-i` *installdir*]	The install directory.

[-q]	Quiet mode. Output only includes tuning recommendations. OS version statements are omitted.
[-v]	Version. Gives the build date identifying the version of the toll.

keyupg *args*

Upgrade the key from Lite to normal (only one way).

The `keyupg` subcommand supports the following arguments:

-k*key*	The key to be upgraded.
-f *key_file_path*	The key file path.

ldif2db-task args

Import data to the directory. It create an entry in the directory that launches this dynamic task. The entry is generated based upon the values you provide for each option. The server must be running when you run this subcommand.

The `ldif2sb-task` subcommand supports the following arguments:

[-c]	Request that only the core db is created without attribute indexes.
-D rootDN	User DN with root permissions, such as Directory Manager.
[-g *string*]	Generation of a unique ID. Enter `none` for no unique ID to be generated and `deterministic` for the generated unique ID to be name-based. Generates a time based unique ID by default. If you use the `deterministic` generation to have a name-based unique ID, you can also

	specify the namespace you want the server to use as follows:
`-g deterministic` *namespace_id*	
	where namespace_id is a string of characters in the following format
`00-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx`	
	Use this option if you want to import the same LDIF file into two different directory servers, and you want the contents of both directories to have the same set of unique IDs. If unique IDs already exist in the LDIF file you are importing, then the existing IDs are imported to the server regardless of the options you have specified.
[`-G` *namespace_id*]	Generate a namespace ID as a name-based unique ID. This is the same as specifying `-g deterministic`.
{`-i` *filename*}*	File name of the input LDIF files. When you import multiple files, they are imported in the order in which you specify them on the command line.
`-j` *file*	Password associated with the user DN. This option allows the password to be stored in clear text in the named file for scripting. This is considered insecure. Use with extreme caution.
`-n` backend_instance	Instance to be imported.

[-O]	Request that only the core db is created without attribute indexes.
{-s *includesuffix* }*	Suffixes to be included. This argument can also be used to specify the subtrees to be included with -n.
-w *password*	Password associated with the user DN. Supplying the password on the command line is visible using the /bin/ps command. This is considered insecure. Use with extreme caution. The value - can be used in place the password. The program prompts the user for a password to be entered from the terminal.
[{-x *excludesuffix* }*]	
[-v]	Verbose mode.

ldif *args*

Format LDIF files, and create base 64 encoded attribute values. With Base 64 Encoding you can represent binary data, such as a JPEG image, in LDIF by using base 64 encoding. You identify base 64 encoded data by using the :: symbol. The ldifsubcommand takes any input and formats it with the correct line continuation and appropriate attribute information. The subcommand also senses whether the input requires base 64 encoding.

The ldif subcommand supports the following arguments

[-b]	Interpret the entire input as a single binary value. If -b is not present, each line is considered to be a separate input value.
[*attrtype*]	If -b is specified, the output is attrtype:: <base 64 encoded

`value`.

ldif2db *args*

Import the data to the directory. To run this subcommand the server must be stopped. Note that ldif2db supports LDIF version 1 specifications. You can load an attribute using the URL specifier notation, for example:
`jpegphoto:file:///tmp/myphoto.jpg`

[`-c`]

Merge chunk size.

[`-g` *string*]

Generation of a unique ID. Type none for no unique ID to be generated and deterministic for the generated unique ID to be name-based. By default a time based unique ID is generated.

If you use the deterministic generation to have a name-based unique ID, you can also specify the namespace you want the server to use as follows:

`-g deterministic` *namespace_id*

where *namespace_id* is a string of characters in the following format:

`00-xxxxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxx`

Use this option if you want to import the same LDIF file into two different directory servers, and you want the contents of both directories to have the same set of unique IDs. If unique IDs already exist in the LDIF file you are importing, then the existing IDs are imported to the server regardless of the options you have specified.

	[-G *naemspace_id*]	Generate a namespace ID as a name-based unique ID. This is the same as specifying the -g deterministic option.
	{- *filename*}*	File name of the input LDIF file(s). When you import multiple files, they are imported in the order in which you specify them on the command line.
	-n *backend_instance*	Instance to be imported.
	[-O]	Request that only the core db is created without attribute indexes.
	{-s *includesuffix*}*	Suffixes to be included or to specify the subtrees to be included if -n has been used.
	[{-x *excludesuffix*}*]	Suffixes to be excluded
ldif2ldap *rootDN password filename*	Perform an import operation over LDAP to the Directory Server. To run this subcommand the server must be running. The ldif2ldap subcommand supports the following arguments:	
	rootdn	User DN with root permissions, such as Directory Manager.
	password	Password associated with the user DN.
	filename	File name of the file to be imported. When you import multiple files, they are imported in the order in which you specify them on the command line.
magt CONFIG INIT	Start SNMP master agent. The Config and INIT files are in /usr/iplanet/ds5/plugins/snmp/magt. For more information, see the 『iPlanet Directory Server 5.1 Administrator's Guide』 . The magt subcommand supports the following options:	

CONFIG — The CONFIG file defines the community and the manager that master agent works with. Specify the manager value as a valid system name or an IP address.

INIT — The INIT file is a nonvolatile file that contains information from the MIB-II system group, including system location and contact information. If INIT doesn't already exist, starting the master agent for the first time creates it. An invalid manager name in the CONFIG file causes the master agent start-up to fail.

monitor — Retrieves performance monitoring information using the `ldapsearch` command-line utility.

mmldif *args* — Combine multiple LDIF files into a single authoritative set of entries. Typically each LDIF file is from a master server cooperating in a multi master replication agreement.[e.g. masters that refuse to sync up for whatever reason]. Optionally, it can generate LDIF change files that could be applied to original to bring it up to date with authoritative. At least two input files must be specified.

The `mmldif` subcommand supports the following arguments:

[`-c` *inputfile* ...] — Write a change file (.delta) for each input file. Specify *inputfile* as the input LDIF files.

[`-D`] — Print debugging information.

[`-o` *out.ldif*] — Write authoritative data to this file.

nativetoascii *args* — Convert one language encoding to another. For example, convert a native language to UTF-8 format.

The `nativetoascii` subcommand supports the following options:

`-d` *Encodings Directory* — Path to the

		directory which contains the `conv` directory
	[`-i` *input_filename* `-o` *output_filename*]	The input file name and output file name.
	`-l`	List supported encodings
	`-r`	Replace existing files.
	`-s` *suffix*	Suffix to be mapped to the backend.
	`-s` *SourceEncoding*	Source Encoding of input stream.
	`-t` *TargetEncoding*	Target Encoding of output stream.
	`-v`	Verbose output.
pwdhash *args*	Print the encrypted form of a password using one of the server's encryption algorithms. If a user cannot log in, you can use this script to compare the user's password to the password stored in the directory. The `pwdhash` subcommand supports the following arguments:	

	-c *comparepwd* | -s *scheme*	The available schemes are SSHA, SHA, CRYPT and CLEARE. It generates the encrypted passwords according to scheme's algorithm. The -c specifies the encrypted password to be compared with. The result of comparison is either OK or doesn't match.
	-D *instance-dir*	The instance directory.
	[-H]	The passwords are hex-encoded.
	password ...	The clear passwords to generate encrypted form from or to be compared with.
restart	Restarts the directory server. When the -s option is not specified, restarts all instances of servers. When the -s option is specified, restarts the server specified by -s. restart-admin Restarts the administration server.	
restoreconfig	Restores the most recently saved Administration Server configuration information to the NetscapeRoot partition under /var/ds5/slapd-serverID/confbak.	
sagt -c CONFIG	Start proxy SNMP agent. For more information, see the 『iPlanet Directory Server 5.1 Administrator's Guide』. The sagt subcommand supports the following options:	

	-c *configfile* The CONFIG file includes the port that the SNMP daemon listens to. It also needs to include the MIB trees and traps that the proxy SNMP agent forwards. Edit the CONFIG file located in /usr/iplanet/ds5/plugins/snmp/sagt.
saveconfig	Saves the administration server configuration information to the /var/ds5/slapd-serverID/confbak directory.
setup [-f *configuration_file*]	Configures an instance of the directory server or administration server. Creates a basic configuration for the directory server and the administrative server that is used to manage the directory. The setup subcommand has two modes of operation. You can invoke it with a curses-based interaction to gather input. Alternatively, you can provide input in a configuration file using the -f option. The setup subcommand supports the following option: -f *configuration_file* Specifies the configuration file for silent installation.
start	Starts the directory server. When the -s option is not specified, starts servers of all instances. When the -s option is specified, starts the server instance specified by -s.
start-admin	Starts the directory server. When the -s option is not specified, restarts all instances of servers. When the -s option is specified, restarts the server specified by -s.
startconsole	Starts the directory console..
stop	Stops the directory server. When the -s option is not specified, restarts all instances of servers. When the -s option is specified, restarts the server specified by -s.
stop-admin	Stop the administration server.

suffix2instance {-s *suffix*}

Map a suffix to a backend name.

Specify -s *suffix* as the suffix to be mapped to the backend.

uninstall

Uninstalls the directory server and the administration server.

This subcommand stops servers of all instances and removes all the changes created by setup.

vlvindex *args*

Create virtual list view (VLV) indexes, known in the Directory Server Console as Browsing Indexes. The server must be stopped beforehand.

The `vlvindex` subcommand supports the following arguments:

-d *debug_level*

Specify the debug level to use during index creation. Debug levels are defined in `nsslapd-errorlog-level (error Log Level)`. See the 『iPlanet Directory Server 5.1 Configuration, Command, and File Reference』.

-n *backend_instance*

Name of the database containing the entries to index.

-s *suffix*

Name of the suffix containing the entries to index.

-T *VLVTag*

Name of the database containing the entries to index.

オプション

Options for the `directoryserver` command itself must appear before the subcommand argument.

The following options are supported:

-s *server-instance*
-server *server-instance*

The server instance name. Specify the directory server instance to process the command against. For some of the listed

subcommands the server instance is optional and for other sub commands it is a required option.

使用例

例 1 Starting All Instances of the Directory Servers

The following command starts all the instances of the directory servers:

```
example% directoryserver start
```

例 2 Starting the Instances of myhost of the Directory Server

The following command starts the instances `myhost` of the directory server.

```
example% directoryserver -s myhost start
```

例 3 Running the Monitor Tool and Outputting the Current Status

The following command runs the monitor tool and output the current status of the `ephesus` directory instance.

```
example% directoryserver -s ephesus monitor
```

例 4 Running the idsktune Tool and Outputting Performance Tuning Information

The following command runs the idsktune tool and outputs performance tuning information:

```
example% directoryserver idsktune
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`non-zero` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	IPLTdsr, IPLTdsu

関連項目

『iPlanet Directory Server 5.1 Administrator's Guide』

『iPlanet Directory Server 5.1 Configuration, Command, and File Reference』

名前

disks – creates /dev entries for hard disks attached to the system

形式

/usr/sbin/disks [-C] [-r *rootdir*]

機能説明

devfsadm(1M) is now the preferred command for /dev and should be used instead of disks.

disks creates symbolic links in the /dev/dsk and /dev/rdsk directories pointing to the actual disk device special files under the /devices directory tree. It performs the following steps:

1. disks searches the kernel device tree to see what hard disks are attached to the system. It notes the /devices pathnames for the slices on the drive and determines the physical component of the corresponding /dev/dsk or /dev/rdsk name.
2. The /dev/dsk and /dev/rdsk directories are checked for disk entries – that is, symbolic links with names of the form c*N*[t*N*]d*N*s*N*, or c*N*[t*N*]dNp*N*, where *N* represents a decimal number. c*N* is the logical controller number, an arbitrary number assigned by this program to designate a particular disk controller. The first controller found on the first occasion this program is run on a system, is assigned number 0. t*N* is the bus-address number of a subsidiary controller attached to a peripheral bus such as SCSI or IPI (the target number for SCSI, and the facility number for IPI controllers). d*N* is the number of the disk attached to the controller. s*N* is the *slice* number on the disk. p*N* is the FDISK partition number used by fdisk(1M). (x86 Only)
3. If only some of the disk entries are found in /dev/dsk for a disk that has been found under the /devices directory tree, disks creates the missing symbolic links. If none of the entries for a particular disk are found in /dev/dsk, disks checks to see if any entries exist for other disks attached to the same controller, and if so, creates new entries using the same controller number as used for other disks on the same controller. If no other /dev/dsk entries are found for slices of disks belonging to the same physical controller as the current disk, disks assigns the lowest-unused controller number and creates entries for the disk slices using this newly-assigned controller number.

disks is run automatically each time a reconfiguration-boot is performed or when add_drv(1M) is executed. When invoking disks manually, first run drvconfig(1M) to ensure /devices is consistent with the current device configuration.

Notice to Driver Writers

disks considers all devices with a node type of DDI_NT_BLOCK, DDI_NT_BLOCK_CHAN, DDI_NT_CD, DDI_NT_BLOCK_WWN or DDI_NT_CD_CHAN to be disk devices. disks requires the minor name of disk devices obey the following format conventions.

The minor name for block interfaces consists of a single lowercase ASCII character, a through u, representing the slices and the primary partitions. The minor name for logical drive block interfaces consists of the strings p5 through p36. The minor name for character (raw) interfaces consists of a single lowercase ASCII character, a through a, followed by the string ,raw, representing the slices and the primary partitions. The minor name for logical drive character (raw) interfaces consists of the string p5 through p36 followed by ,raw.

disks performs the following translations:

- a through p to s0 through s15
- q through u to p0 through p4
- p5 through p36 to p5 through p36

SPARC drivers should only use the first eight slices: a through h, while x86 drivers can use a through u, with q through u corresponding to fdisk(1M) primary partitions. q represents the entire disk, while r, s, t, and u represent up to four additional primary partitions. For logical drives, p5 to p36 correspond to the 32 logical drives that are supported. The device nodes for logical drives change dynamically as and when they are created or deleted.

To prevent disks from attempting to automatically generate links for a device, drivers must specify a private node type and refrain from using a node type: DDI_NT_BLOCK, DDI_NT_BLOCK_CHAN, DDI_NT_CD, or DDI_NT_CD_CHAN when calling ddi_create_minor_node(9F).

オプション

The following options are supported:

-C
: Causes disks to remove any invalid links after adding any new entries to /dev/dsk and /dev/rdsk. Invalid links are links which refer to non-existent disk nodes that have been removed, powered off, or are otherwise inaccessible.

-r *rootdir*
: Causes disks to presume that the /dev/dsk, /dev/rdsk and /devices directory trees are found under *rootdir*, not directly under /.

エラー

If disks finds entries of a particular logical controller linked to different physical controllers, it prints an error message and exits without making any changes to the /dev directory, since it cannot determine which of the two alternative logical-to-physical mappings is correct. The links should be manually corrected or removed before another reconfiguration-boot is performed.

使用例

例 1 Creating Block and Character Minor Devices

The following example demonstrates creating the block and character minor devices from within the xkdisk driver's attach(9E) function.

```
#include    <sys/dkio.h>
/*
 * Create the minor number by combining the instance number
 * with the slice number.
 */
#define MINOR_NUM(i, s)   ((i) << 4 | (s))

int
xkdiskattach(dev_info_t *dip, ddi_attach_cmd_t cmd)
{
    int instance, slice;
    char name[8];
```

例 1 Creating Block and Character Minor Devices (続き)

```
        /* other stuff in attach... */

    instance = ddi_get_instance(dip);
    for (slice = 0; slice < V_NUMPAR; slice++) {
        /*
         * create block device interface
         */
        sprintf(name, "%c", slice + 'a');
        ddi_create_minor_node(dip, name, S_IFBLK,
            MINOR_NUM(instance, slice), DDI_NT_BLOCK_CHAN, 0);

        /*
         * create the raw (character) device interface
         */
        sprintf(name,"%c,raw", slice + 'a');
        ddi_create_minor_node(dip, name, S_IFCHR,
            MINOR_NUM(instance, slice), DDI_NT_BLOCK_CHAN, 0);
    }
}
```

Installing the `xkdisk` disk driver on a Sun Fire 4800, with the driver controlling a SCSI disk (target 3 attached to an `isp(7D)` SCSI HBA) and performing a reconfiguration-boot (causing disks to be run) creates the following special files in `/devices`.

```
# ls -l /devices/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4/
brw-r-----   1 root sys   32, 16 Aug 29 00:02 xkdisk@3,0:a
crw-r-----   1 root sys   32, 16 Aug 29 00:02 xkdisk@3,0:a,raw
brw-r-----   1 root sys   32, 17 Aug 29 00:02 xkdisk@3,0:b
crw-r-----   1 root sys   32, 17 Aug 29 00:02 xkdisk@3,0:b,raw
brw-r-----   1 root sys   32, 18 Aug 29 00:02 xkdisk@3,0:c
crw-r-----   1 root sys   32, 18 Aug 29 00:02 xkdisk@3,0:c,raw
brw-r-----   1 root sys   32, 19 Aug 29 00:02 xkdisk@3,0:d
crw-r-----   1 root sys   32, 19 Aug 29 00:02 xkdisk@3,0:d,raw
brw-r-----   1 root sys   32, 20 Aug 29 00:02 xkdisk@3,0:e
crw-r-----   1 root sys   32, 20 Aug 29 00:02 xkdisk@3,0:e,raw
brw-r-----   1 root sys   32, 21 Aug 29 00:02 xkdisk@3,0:f
crw-r-----   1 root sys   32, 21 Aug 29 00:02 xkdisk@3,0:f,raw
brw-r-----   1 root sys   32, 22 Aug 29 00:02 xkdisk@3,0:g
crw-r-----   1 root sys   32, 22 Aug 29 00:02 xkdisk@3,0:g,raw
brw-r-----   1 root sys   32, 23 Aug 29 00:02 xkdisk@3,0:h
crw-r-----   1 root sys   32, 23 Aug 29 00:02 xkdisk@3,0:h,raw
```

/dev/dsk will contain the disk entries to the block device nodes in `/devices`

```
# ls -l /dev/dsk
/dev/dsk/c0t3d0s0 -> ../../devices/[...]/xkdisk@3,0:a
```

例 1 Creating Block and Character Minor Devices (続き)

```
/dev/dsk/c0t3d0s1 -> ../../devices/[...]/xkdisk@3,0:b
/dev/dsk/c0t3d0s2 -> ../../devices/[...]/xkdisk@3,0:c
/dev/dsk/c0t3d0s3 -> ../../devices/[...]/xkdisk@3,0:d
/dev/dsk/c0t3d0s4 -> ../../devices/[...]/xkdisk@3,0:e
/dev/dsk/c0t3d0s5 -> ../../devices/[...]/xkdisk@3,0:f
/dev/dsk/c0t3d0s6 -> ../../devices/[...]/xkdisk@3,0:g
/dev/dsk/c0t3d0s7 -> ../../devices/[...]/xkdisk@3,0:h
```

and /dev/rdsk will contain the disk entries for the character device nodes in `/devices`

```
# ls -l /dev/rdsk
/dev/rdsk/c0t3d0s0 -> ../../devices/[...]/xkdisk@3,0:a,raw
/dev/rdsk/c0t3d0s1 -> ../../devices/[...]/xkdisk@3,0:b,raw
/dev/rdsk/c0t3d0s2 -> ../../devices/[...]/xkdisk@3,0:c,raw
/dev/rdsk/c0t3d0s3 -> ../../devices/[...]/xkdisk@3,0:d,raw
/dev/rdsk/c0t3d0s4 -> ../../devices/[...]/xkdisk@3,0:e,raw
/dev/rdsk/c0t3d0s5 -> ../../devices/[...]/xkdisk@3,0:f,raw
/dev/rdsk/c0t3d0s6 -> ../../devices/[...]/xkdisk@3,0:g,raw
/dev/rdsk/c0t3d0s7 -> ../../devices/[...]/xkdisk@3,0:h,raw
```

ファイル

`/dev/dsk/*` Disk entries (block device interface)

`/dev/rdsk/*` Disk entries (character device interface)

`/devices/*` Device special files (minor device nodes)

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`add_drv(1M)`, `devfsadm(1M)`, `fdisk(1M)`, `attributes(5)`, `isp(7D)`, `devfs(7FS)`, `dkio(7I)`, `attach(9E)`, `ddi_create_minor_node(9F)`

『Writing Device Drivers』

使用上の留意点

`disks` silently ignores malformed minor device names.

名前

diskscan – perform surface analysis

形式

`diskscan [-W] [-n] [-y] raw_device`

機能説明

`diskscan` is used by the system administrator to perform surface analysis on a portion of a hard disk. The disk portion may be a raw partition or slice; it is identified using its raw device name. By default, the specified portion of the disk is read (non-destructive) and errors reported on standard error. In addition, a progress report is printed on standard out. The list of bad blocks should be saved in a file and later fed into `addbadsec(1M)`, which will remap them.

オプション

The following options are supported:

`-n` Causes `diskscan` to suppress linefeeds when printing progress information on standard out.

`-W` Causes `diskscan` to perform write and read surface analysis. This type of surface analysis is destructive and should be invoked with caution.

`-y` Causes `diskscan` to suppress the warning regarding destruction of existing data that is issued when `-W` is used.

オペランド

The following operands are supported:

raw_device The address of the disk drive (see `FILES`).

ファイル

The raw device should be `/dev/rdsk/c?[t?]d?[ps]?`. See `disks(1M)` for an explanation of SCSI and IDE device naming conventions.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86
Availability	system/core-os

関連項目

`addbadsec(1M)`, `disks(1M)`, `fdisk(1M)`, `fmthard(1M)`, `format(1M)`, `attributes(5)`

注意事項

The `format(1M)` utility is available to format, label, analyze, and repair SCSI disks. This utility is included with the `diskscan`, `addbadsec(1M)`, `fdisk(1M)`, and `fmthard(1M)` commands available for x86. To format an IDE disk, use the DOS `format` utility; however, to label, analyze, or repair IDE disks on x86 systems, use the Solaris `format(1M)` utility.

名前

dispadmin – process scheduler administration

形式

```
dispadmin -l
dispadmin -c class {-g [-r res] | -s file}
dispadmin -d [class]
```

機能説明

The `dispadmin` command displays or changes process scheduler parameters while the system is running.

`dispadmin` does limited checking on the values supplied in *file* to verify that they are within their required bounds. The checking, however, does not attempt to analyze the effect that the new values have on the performance of the system. Inappropriate values can have a negative effect on system performance. (See 『Oracle Solaris の管理: 一般的なタスク』.)

オプション

The following options are supported:

`-c` *class*
: Specifies the class whose parameters are to be displayed or changed. Valid *class* values are: `RT` for the real-time class, `TS` for the time-sharing class, `IA` for the inter-active class, `FSS` for the fair-share class, and FX for the fixed-priority class. The time-sharing and inter-active classes share the same scheduler, so changes to the scheduling parameters of one will change those of the other.

`-d` [*class*]
: Sets or displays the name of the default scheduling class to be used on reboot when starting `svc:/system/scheduler:default`. If class name is not specified, the name and description of the current default scheduling class is displayed. If class name is specified and is a valid scheduling class name, then it is saved in `dispadmin`'s private configuration file `/etc/dispadmin.conf`. Only super-users can set the default scheduling class.

`-g`
: Gets the parameters for the specified class and writes them to the standard output. Parameters for the real-time class are described in `rt_dptbl(4)`. Parameters for the time-sharing and inter-active classes are described in `ts_dptbl(4)`. Parameters for the fair-share class are described in `FSS(7)`. Parameters for the fixed-priority class are described in `fx_dptbl(4)`.

  The `-g` and `-s` options are mutually exclusive: you may not retrieve the table at the same time you are overwriting it.

`-l`
: Lists the scheduler classes currently configured in the system.

`-r` *res*
: When using the `-g` option you may also use the `-r` option to specify a resolution to be used for outputting the time quantum values. If no resolution is specified, time quantum values are in milliseconds. If *res* is specified it must be a positive integer between 1 and 1000000000 inclusive, and the resolution used is the reciprocal of *res* in seconds. For

example, a *res* value of 10 yields time quantum values expressed in tenths of a second; a *res* value of 1000000 yields time quantum values expressed in microseconds. If the time quantum cannot be expressed as an integer in the specified resolution, it is rounded up to the next integral multiple of the specified resolution.

`-s` *file*

Sets scheduler parameters for the specified class using the values in *file*. These values overwrite the current values in memory—they become the parameters that control scheduling of processes in the specified class. The values in *file* must be in the format output by the `-g` option. Moreover, the values must describe a table that is the same size (has same number of priority levels) as the table being overwritten. Super-user privileges are required in order to use the `-s` option.

Specify time quantum values for scheduling classes in system clock ticks, and not in constant-time units. Time quantum values are based on the value of the kernel's `hz` variable. If kernel variable `hires_tick` is set to `1` to get higher resolution clock behavior, the actual time quanta will be reduced by the order of `10`.

The `-g` and `-s` options are mutually exclusive: you may not retrieve the table at the same time you are overwriting it.

使用例

例 1 Retrieving the Current Scheduler Parameters for the real-time class

The following command retrieves the current scheduler parameters for the real-time class from kernel memory and writes them to the standard output. Time quantum values are in microseconds.

```
dispadmin -c RT -g -r 1000000
```

例 2 Overwriting the Current Scheduler Parameters for the Real-time Class

The following command overwrites the current scheduler parameters for the real-time class with the values specified in `rt.config`.

```
dispadmin -c RT -s rt.config
```

例 3 Retrieving the Current Scheduler Parameters for the Time-sharing Class

The following command retrieves the current scheduler parameters for the time-sharing class from kernel memory and writes them to the standard output. Time quantum values are in nanoseconds.

```
dispadmin -c TS -g -r 1000000000
```

例 4 Overwriting the Current Scheduler Parameters for the Time-sharing Class

The following command overwrites the current scheduler parameters for the time-sharing class with the values specified in `ts.config`.

```
dispadmin -c TS -s ts.config
```

ファイル

/etc/dispadmin.conf
Possible location for argument to -s option.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

priocntl(1), svcs(1), svcadm(1M), priocntl(2), fx_dptbl(4), rt_dptbl(4), ts_dptbl(4), attributes(5), smf(5), FSS(7)

『Oracle Solaris の管理: 一般的なタスク』 『Programming Interfaces Guide』

診断

dispadmin prints an appropriate diagnostic message if it fails to overwrite the current scheduler parameters due to lack of required permissions or a problem with the specified input file.

注意事項

The default scheduling class setting facility is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/scheduler:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Note that disabling the service while it is running will not change anything. The service's status can be queried using the svcs(1) command.

名前

dladm – データリンクの管理

形式

```
dladm show-link [-PZ] [-s [-i interval]] [[-p] -o field[,...]]
     [-z zone[,...]] [link]
dladm rename-link [-R root-dir] link new-link

dladm delete-phys phys-link
dladm show-phys [-PZ] [-Lm] [[-p] -o field[,...]] [-H]
     [-z zone[,...]] [phys-link]

dladm create-aggr [-t] [-R root-dir] [-P policy] [-L mode]
     [-T time] [-u address] -l ether-link1 [-l ether-link2...] aggr-link
dladm modify-aggr [-t] [-R root-dir] [-P policy] [-L mode]
     [-T time] [-u address] aggr-link
dladm delete-aggr [-t] [-R root-dir] aggr-link
dladm add-aggr [-t] [-R root-dir] -l ether-link1 [-l ether-link2...]
     aggr-link
dladm remove-aggr [-t] [-R root-dir] -l ether-link1 [-l ether-link2...]
     aggr-link
dladm show-aggr [-PLxZ] [-s [-i interval]] [[-p] -o field[,...]]
     [-z zone[,...]] [aggr-link]

dladm create-bridge [-P protect] [-R root-dir] [-p priority]
     [-m max-age] [-h hello-time] [-d forward-delay] [-f force-protocol]
     [-l link...] bridge-name
dladm modify-bridge [-P protect] [-R root-dir] [-p priority]
     [-m max-age] [-h hello-time] [-d forward-delay] [-f force-protocol]
     bridge-name
dladm delete-bridge [-R root-dir] bridge-name
dladm add-bridge [-R root-dir] -l link [-l link...]bridge-name
dladm remove-bridge [-R root-dir] -l link [-l link...] bridge-name
dladm show-bridge [-flt] [-s [-i interval]] [[-p] -o field,...]
     [bridge-name]

dladm create-vlan [-ft] [-R root-dir] -l ether-link -v vid [vlan-link]
dladm delete-vlan [-t] [-R root-dir] vlan-link
dladm show-vlan [-PZ] [[-p] -o field[,...]] [-z zone[,...]] [vlan-link]

dladm scan-wifi [[-p] -o field[,...]] [wifi-link]
dladm connect-wifi [-e essid] [-i bssid] [-k key,...]
     [-s none | wep | wpa ] [-a open | shared] [-b bss | ibss] [-c]
     [-m a | b | g | n ] [-T time] [wifi-link]
dladm disconnect-wifi [-a] [wifi-link]
dladm show-wifi [-Z] [[-p] -o field[,...]] [-z zone[,...]] [wifi-link]

dladm show-ether [-xZ] [[-p] -o field[,...]] [-z zone[,...]] [ether-link]

dladm set-linkprop [-t] [-R root-dir] -p prop=value[,...] link
dladm reset-linkprop [-t] [-R root-dir] [-p prop[,...]] link
dladm show-linkprop [-PZ] [[-c] -o field[,...]] [-p prop[,...]]
     [-z zone[,...]] [link]
```

```
dladm create-secobj [-t] [-R root-dir] [-f file] -c class secobj
dladm delete-secobj [-t] [-R root-dir] secobj[,...]
dladm show-secobj [-P] [[-p] -o field[,...]] [secobj,...]

dladm create-vnic [-t] -l link [-R root-dir] [-m value | auto |
     {factory [-n slot-identifier]} | {vrrp -A {inet | inet6} -V vrid}
     | {random [-r prefix]}] [-v vlan-id] [-p prop=value[,...]] vnic-link
dladm delete-vnic [-t] [-R root-dir] vnic-link
dladm show-vnic [-pPZ] [-s [-i interval]] [-o field[,...]]
     [-l link] [-z zone[,...]] [vnic-link]

dladm create-etherstub [-t] [-R root-dir] etherstub
dladm delete-etherstub [-t] [-R root-dir] etherstub
dladm show-etherstub [-Z] [-z zone[,...]] [etherstub]

dladm create-iptun [-t] [-R root-dir] -T type [-a {local|remote}=addr,...]
     iptun-link
dladm modify-iptun [-t] [-R root-dir] -a {local|remote}=addr,...
     iptun-link
dladm delete-iptun [-t] [-R root-dir] iptun-link
dladm show-iptun [-PZ] [[-p] -o field[,...]] [-z zone[,...]] [iptun-link]

dladm create-part [-t] [-f] -l ib-link [-R root-dir] -P pkey
     [-p prop=value[,...]] part-link
dladm delete-part [-t] [-R root-dir] part-link
dladm show-part [-pP] [-o field[,...]] [-l ib-link]  [part-link]

dladm show-ib [-pP] [-o field[,...]] [ib-link]

dladm show-usage [-a] -f filename [-p plotfile -F format] [-s time]
     [-e time] [link]
```

機能説明

dladm コマンドは、データリンクを管理するために使用されます。データリンクはシステム内で、TCP/IP などのプロトコルスタックの下で plumb できる STREAMS DLPI (v2) インタフェースとして表されます。各データリンクは、単一のネットワークデバイスまたはデバイスの集積体を使用して、ネットワーク間でパケットを送受信します。

各 dladm サブコマンドは、次のいずれかのオブジェクトに対して動作します。

link
: データリンク。名前で識別されます。一般に、この名前には任意の英数字 (および、下線 _ またはピリオド .) を使用できますが、英字で始まり、数字で終わる必要があります。データリンク名には最大 31 文字を使用でき、最後の数字は 0 - 4294967294 の間の値にする必要があります。最後の数字を 0 で始めてはいけません。データリンク名は 3 - 8 文字にすることをお勧めします。

一部のサブコマンドは、データリンクの特定のタイプまたはクラスに対してのみ動作します。その場合は、次のオブジェクト名が使用されます。

aggr-link
: 集約データリンク (またはキー、「注意事項」を参照)。

`ether-link`
物理 Ethernet データリンク。

`iptun-link`
IP トンネルリンク。

`part-link`
InfiniBand (IB) パーティションデータリンク。

`phys-link`
物理データリンク。

`vlan-link`
VLAN データリンク。

`vnic-link`
リンクまたは `etherstub` 上に作成された仮想ネットワークインタフェース。これは、マシン上でネットワークインタフェースカードであるかのように扱うことが可能な疑似デバイスです。

`wifi-link`
WiFi データリンク。

`bridge`
ブリッジインスタンス。管理用に選択された名前で識別されます。この名前には任意の英数字または下線 _ を使用できますが、英字で始まり、英字で終わる必要があります。ブリッジ名には最大 31 文字を使用できます。`default` という名前は、`SUNW` で始まるすべての名前を表すため、予約されています。

ブリッジ名にゼロ (`0`) を付加すると、可観測性に使用される有効なリンク名が生成されることに注意してください。

また、下の `dladm` サブコマンドで説明されているブリッジ関連のサブコマンドには `pkg://solaris/network/bridging` パッケージのインストールが必要であることに注意してください。

`dev`
ネットワークデバイス。ドライバ名とインスタンス番号の連結で識別されます。

`etherstub`
物理 NIC の代わりに Ethernet スタブを使用して VNIC を作成することもできます。`etherstub` 上に作成された VNIC は、仮想スイッチを経由して接続されているように表示されるため、物理ハードウェアなしで完全な仮想ネットワークを構築できます。

`part`
IB 物理リンク上に作成された IB パーティションリンク。

`secobj`
セキュリティー保護されたオブジェクト。管理用に選択された名前で識別されます。この名前には、任意の英数字のほか、下線 (_)、ピリオド (.)、およびハイフ

ン (-) を使用できます。セキュリティー保護されたオブジェクトの名前には最大 32 文字を使用できます。

オプション

各 dladm サブコマンドには、独自のオプションセットがあります。ただし、多くのサブコマンドには次の共通オプションがあります。

-R *root-dir*, --root-dir=*root-dir*
作成、削除、名前変更などの操作が適用する代替ルートディレクトリを指定します。

サブコマンド

次のサブコマンドがサポートされています。

dladm show-link [-PZ] [-s [-i *interval*]] [[-p] -o *field*[,...]] [-z *zone*[,...]] [*link*]
すべてのデータリンクまたは指定されたリンクのリンク構成情報を表示します。デフォルトでは、システムは、既知のネットワークデバイスごとに 1 つのデータリンクで構成されます。リンクの統計情報を出力するためのオプションは dlstat(1M) に移行されました。

-o *field*[,...], --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。-s オプション (下記参照) によって変更されていない場合、フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、show-link はすべてのフィールドを表示します。

LINK
データリンクの名前。

ZONE
データリンクの現在のゾーン。

CLASS
データリンクのクラス。dladm は、次のクラスを識別します。

aggr
IEEE 802.3ad リンク集積体。show-aggr サブコマンドは、このクラスのデータリンクに関する詳細を表示します。

bridge
ブリッジインスタンス。管理用に選択された名前で識別されます。

etherstub
etherstub のインスタンス。物理 NIC の代わりに Ethernet スタブを使用して VNIC を作成することもできます。etherstub 上に作成された VNIC は、仮想スイッチを経由して接続されているように表示されるため、物理ハードウェアなしで完全な仮想ネットワークを構築できます。

iptun
IP トンネルリンクのインスタンス。

part
: IP-over-IB インタフェース。show-part サブコマンドは、このクラスのデータリンクに関する詳細を表示します。

phys
: 物理データリンク。show-phys サブコマンドは、このクラスのデータリンクに関する詳細を表示します。

vlan
: VLAN データリンク。show-vlan サブコマンドは、このクラスのデータリンクに関する詳細を表示します。

vnic
: 仮想ネットワークインタフェース。show-vnic サブコマンドは、このクラスのデータリンクに関する詳細を表示します。

MTU
: 表示されるデータリンクの最大送信単位のサイズ。

STATE
: データリンクのリンク状態。この状態は、up、down、unknown のいずれかになります。

BRIDGE
: このリンクが割り当てられているブリッジの名前 (ある場合)。

OVER
: データリンクが動作している物理データリンク。これは、データリンクの aggr、bridge、vlan、および part パーティションクラスに適用されます。1 つの VLAN または IB パーティションは 1 つの物理的なデータリンク上に作成され、1 つのブリッジには複数のリンクが接続され、1 つの集積体は 1 つ以上の物理的なデータリンクで構成されます。

-o オプションが、リンクの統計情報を表示するために使用される -s オプションと組み合わせて使用する場合、フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。

LINK
: データリンクの名前。

IPACKETS
: このリンク上で受信されたパケット数。

RBYTES
: このリンク上で受信されたバイト数。

IERRORS
: 入力エラーの数。

OPACKETS
このリンク上で送信されたパケット数。

OBYTES
このリンク上で送信されたバイト数。

OERRORS
出力エラーの数。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
永続リンクの構成を表示します。

-s, --statistics
リンクの統計情報を表示します。このオプションは dlstat(1M) によって廃止されています。

-i *interval*, --interval=*interval*
統計情報が表示される間隔 (秒) を指定するために -s オプションとともに使用されます。このオプションは dlstat(1M) によって廃止されています。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
指定されたゾーンからのリンクを表示します。デフォルトでは、大域ゾーンから実行されている場合、dladm はすべてのゾーン内のリンクを表示します。ほかのゾーン内のリンクは、対応するゾーン名がその接頭辞として付き、そのあとにスラッシュ (/) 区切り文字が続いて表示されます。例:

```
zone1/net0
```

非大域ゾーンから実行されている場合、このサブコマンドはそのゾーンのリンクのみを表示します。非大域ゾーンでは、ほかのゾーン内のリンクを表示することはできません。

dladm rename-link [-R *root-dir*] *link new-link*
link の名前を *new-link* に変更します。これは、リンクに意味のある名前を付けたり、削除されたデバイスのリンクプロパティーのような既存のリンク構成を新しいデバイスに関連付けたりするために使用されます。このサブコマンドの使用方法の具体例については、「使用例」の節を参照してください。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

`dladm delete-phys` *phys-link*
このコマンドは、システムから削除された物理ハードウェアに関連付けられているリンクの永続構成を削除するために使用されます。「使用例」の節を参照してください。

`dladm show-phys` [`-PZ`] [`-Lm`] [[`-p`] `-o` *field*[,...]] [`-H`] [`-z` *zone*[,...]] [*phys-link*]
物理デバイスと、すべての物理リンクまたは指定された物理リンクの属性を表示します。`-P` を指定しない場合は、実行中のシステム上で使用可能な物理リンクのみが表示されます。

`-H`
NIC ドライバによって返される、ハードウェアリソースの使用状況を表示します。`-H` からの出力には、次の要素が表示されます。

`LINK`
NIC ドライバに対応する物理デバイス。

`RINGTYPE`
リングのタイプ (RX または TX のどちらか)。

`RINGS`
リングのインデックス。リングとはハードウェアリソースのことで、通常は DMA チャネルにマップされ、特定の用途のためにプログラムできます。たとえば、RX リングをプログラムすると、特定の MAC アドレスに属するパケットのみを受信できます。

`CLIENTS`
リングを使用している MAC クライアント。

`-L`
物理デバイス/リンクの位置情報を表示します。出力は配置順 (つまり、オンボードデバイス、拡張スロットの順) で行われ、位置情報 (PCIexp Slot 2、MB など) は利用可能な場合に提供されます。`-L` からの出力は、次の要素をサポートしています。

`LINK`
NIC ドライバに対応する物理デバイス。

`DEVICE`
このリンクの下にある物理デバイスの名前。

`LOC`
物理的な位置を記述する文字列 (利用可能な場合)。

`-m`
ファクトリ MAC アドレス、そのスロット識別子、およびその可用性を一覧表示します。

`-o` *field*, `--output=`*field*
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべての

フィールドを表示するための特殊な値 all にする必要があります。リンクごとに、次のフィールドを表示できます。

LINK
データリンクの名前。

MEDIA
物理データリンクによって提供されるメディアタイプ。

STATE
リンクの状態。この状態は、up、down、unknown のいずれかになります。

SPEED
リンクの現在の速度 (Mbps 単位)。

DUPLEX
Ethernet リンクでリンクステータスが up の場合は、そのリンクの二重通信のステータス (全二重/半二重) が表示されます。それ以外の場合は、二重通信の状態は unknown と表示されます。

DEVICE
このリンクの下にある物理デバイスの名前。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
このオプションを指定すると、システムから削除されたリンクを含むすべてのリンクの永続構成が表示されます。出力には FLAGS 列が表示されます。この列の r フラグは、物理リンクに関連付けられた物理デバイスが削除されたことを示します。このようなリンクの場合は、delete-phys を使用すると、そのリンクの構成をシステムから消去できます。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm create-aggr [-t] [-R *root-dir*] [-P *policy*] [-L *mode*] [-T *time*] [-u *address*] -l *ether-link1* [-l *ether-link2*...] *aggr-link*
一連のリンクを、*aggr-link* という名前の 1 つの IEEE 802.3ad リンク集積体と組み合わせます。また、下位互換性のために、整数キーによる集積体用リンク名の生成もサポートされています。下記の * -aggr サブコマンドの多くでは、キーによる特定の集積体の参照もサポートされていますが、集約リンク名の使用をお勧めします。キーの詳細については、「注意事項」の節を参照してください。

dladm は、ポート集約のためのポート選択ポリシーをいくつかサポートしています。(下記の -P オプションの説明を参照してください。)ポリシーを指定しない場合、create-aggr は、-P オプションで説明されているデフォルトの L4 ポリシーを使用します。

-l *ether-link*, --link=*ether-link*
集積体の各 Ethernet リンク (またはポート) は、-l オプションのあとに、その集積体に含めるリンクの名前を使用して指定します。複数の -l オプションを指定すると、複数のリンクが集積体に含まれます。旧バージョンの Solaris との下位互換性を確保するため、dladm コマンドでは、-d オプション (または --dev) をデバイス名とともに使用して、基本となるデバイス名でリンクを指定する方法もサポートされています。また、-l オプションを取るほかの *-aggr サブコマンドでも -d を使用できます。

-t, --temporary
集積体を一時的なものとして指定します。一時的な集積体は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-P *policy*, --policy=*policy*
送信トラフィックの負荷分散に使用するポート選択ポリシーを指定します。このポリシーでは、パケットを送信するためにどの *dev* オブジェクトが使用されるかを指定します。ポリシーは、1 つ以上のレイヤー指定子をコンマで区切ったリストです。レイヤー指定子は、次のいずれかです。

L2
パケットの発信元および着信先 MAC アドレスに従って送信デバイスを選択します。

L3
パケットの発信元および着信先 IP アドレスに従って送信デバイスを選択します。

L4
パケットに含まれている上位レイヤーのプロトコル情報に従って送信デバイスを選択します。TCP と UDP の場合、これには発信元および着信先ポートが含まれます。IPsec の場合、これには SPI (セキュリティーパラメータインデックス) が含まれます。

たとえば、上位レイヤーのプロトコル情報を使用するには、次のポリシーを使用できます。

```
-P L4
```

ポリシー L4 がデフォルトであることに注意してください。

発信元および着信先 MAC アドレスとともに発信元および着信先 IP アドレスを使用するには、次のポリシーを使用できます。

```
-P L2,L3
```

-L *mode*, --lacp-mode=*mode*
LACP を使用するかどうか、および使用する場合は LACP が動作するモードを指定します。サポートされている値は、off、active、または passive です。

-T *time*, --lacp-timer=*time*
LACP タイマー値を指定します。サポートされている値は、short または long です。

-u *address*, --unicast=*address*
集積体に使用される固定のユニキャストハードウェアアドレスを指定します。このオプションが指定されていない場合、アドレスはコンポーネントデバイスのアドレスのセットから自動的に選択されます。

dladm modify-aggr [-t] [-R *root-dir*] [-P *policy*] [-L *mode*] [-T *time*] [-u *address*] *aggr-link*
指定された集積体のパラメータを変更します。

-t, --temporary
変更を一時的なものとして指定します。一時的な集積体は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-P *policy*, --policy=*policy*
送信トラフィックの負荷分散に使用するポート選択ポリシーを指定します。有効なポリシー値については、dladm create-aggr を参照してください。

-L *mode*, --lacp-mode=*mode*
LACP を使用するかどうか、および使用する場合は LACP が動作するモードを指定します。サポートされている値は、off、active、または passive です。

-T *time*, --lacp-timer=*time*
LACP タイマー値を指定します。サポートされている値は、short または long です。

-u *address*, --unicast=*address*
集積体に使用される固定のユニキャストハードウェアアドレスを指定します。このオプションが指定されていない場合、アドレスはコンポーネントデバイスのアドレスのセットから自動的に選択されます。

(固定のユニキャストハードウェアアドレスを変更すると、集積体のために以前に定義された mac-address リンクプロパティーがすべて上書きされることに注意してください。「一般的なリンクのプロパティー」を参照してください。)

dladm delete-aggr [-t] [-R *root-dir*] *aggr-link*
指定された集積体を削除します。

`-t`, `--temporary`
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`dladm add-aggr` [`-t`] [`-R` *root-dir*] `-l` *ether-link1* [`--link=`*ether-link2*...] *aggr-link*
指定された集積体にリンクを追加します。

`-l` *ether-link*, `--link=`*ether-link*
集積体に追加する Ethernet リンクを指定します。複数の `-l` オプションを指定すると、複数のリンクを追加できます。

`-t`, `--temporary`
追加を一時的なものとして指定します。一時的な追加は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`dladm remove-aggr` [`-t`] [`-R` *root-dir*] `-l` *ether-link1* [`--l=`*ether-link2*...] *aggr-link*
指定された集積体からリンクを削除します。

`-l` *ether-link*, `--link=`*ether-link*
集積体から削除する Ethernet リンクを指定します。複数の `-l` オプションを指定すると、複数のリンクを追加できます。

`-t`, `--temporary`
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`dladm show-aggr` [`-PLxZ`] [`-s` [`-i` *interval*]] [[`-p`] `-o` *field*[,...]] [`-z` *zone*[,...]] [*aggr-link*]
すべての集積体または指定された集積体の集積構成 (デフォルト) または LACP 情報を表示します。

デフォルトでは (オプションの指定なし)、次のフィールドを表示できます。

`LINK`
集約リンクの名前。

`POLICY`
集積体の LACP ポリシー。指定可能な値については、`create-aggr` の `-P` オプションを参照してください。

ADDRPOLICY
: 集積体がユニキャスト MAC アドレスを自動的に構成するように構成されている場合は auto (集積体を作成または変更するために -u オプションが使用されなかった場合のデフォルトです)、-u を使用して固定の MAC アドレスが設定された場合は fixed になります。

LACPACTIVITY
: 集積体の LACP モード。指定可能な値は、create-aggr または modify-aggr の -l オプションで設定された、off、active、または passive です。

LACPTIMER
: create-aggr または modify-aggr の -T オプションで設定された、集積体の LACP タイマー値。

FLAGS
: 集積体に関連付けられた状態フラグのセット。指定可能なフラグは f のみです。このフラグは、管理者が create-aggr の -f オプションを使用して集積体を強制的に作成した場合に表示されます。将来、ほかのフラグが定義される可能性があります。

show-aggr コマンドで使用できるオプションは、次のとおりです。

-L, --lacp
: 集約リンクおよび基本となる各ポートの詳細な LACP 情報を表示します。このオプションによって表示される状態情報のほとんどは、IEEE 802.3 で定義されています。このオプションを使用すると、次のフィールドを表示できます。

  LINK
  : 集約リンクの名前。

  PORT
  : 基本となるいずれかの集約ポートの名前。

  AGGREGATABLE
  : 集積体にポートを追加できるかどうか。

  SYNC
  : yes の場合、システムはこのポートが同期されていて、集積体の一部であるとみなします。

  COLL
  : yes の場合、関連付けられたポート上で受信フレームの収集が有効になっています。

  DIST
  : yes の場合、関連付けられたポート上で送信フレームの配信が有効になっています。

DEFAULTED
yes の場合、ポートはデフォルトのパートナー情報を使用しています (つまり、LACP パートナーから LACP データを受信していません)。

EXPIRED
yes の場合、ポートの受信状態が EXPIRED 状態になっています。

-x, --extended
基本となる各ポートに関する詳細情報を含む追加の集積体情報を表示します。-x を使用すると、次のフィールドを表示できます。

LINK
集約リンクの名前。

PORT
基本となるいずれかの集約ポートの名前。

SPEED
リンクまたはポートの速度 (Mbps 単位)。

DUPLEX
リンクステータスが up の場合は、そのリンクまたはポートの二重通信のステータス (全二重/半二重) のステータスが表示されます。それ以外の場合は、二重通信のステータスは unknown と表示されます。

STATE
リンク状態。この状態は、up、down、unknown のいずれかになります。

ADDRESS
リンクまたはポートの MAC アドレス。

PORTSTATE
これは、個々の集約ポートが standby または attached のどちらの状態にあるかを示します。

-o *field*[,...], --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、前述のフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。-o オプションに適用できるフィールドは、各出力モードの下に一覧表示されているフィールドに制限されます。たとえば、-L を使用している場合は、前述のように、-L の下に一覧表示されているフィールドのみを -o とともに使用できます。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
実行中のシステムの状態ではなく、永続的な集積構成を表示します。

-s, --statistics
集積体の統計情報を表示します。このオプションは dlstat(1M) によって廃止されています。

-i *interval*, --interval=*interval*
統計情報が表示される間隔 (秒) を指定するために -s オプションとともに使用されます。このオプションは dlstat(1M) によって廃止されています。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm create-bridge [-P *protect*] [-R *root-dir*] [-p *priority*] [-m *max-age*] [-h *hello-time*] [-d *forward-delay*] [-f *force-protocol*] [-l *link*...] *bridge-name*
802.1D ブリッジインスタンスを作成し、必要に応じて新しいブリッジに 1 つ以上のネットワークリンクを割り当てます。デフォルトでは、システム上にブリッジインスタンスは存在しません。

リンクをブリッジするには、少なくとも 1 つのブリッジインスタンスを作成する必要があります。各ブリッジインスタンスは独立しており、ブリッジ間の転送接続は存在しません。

ブリッジ関連のサブコマンド (特に create-bridge) には pkg://solaris/network/bridging パッケージのインストールが必要であることに注意してください。

-P *protect*, --protect=*protect*
保護の方式を指定します。定義されている保護方式は、Spanning Tree Protocol の stp と、RBridge 上で使用される TRILL の trill です。デフォルト値は stp です。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-p *priority*, --priority=*priority*
ブリッジの優先度を指定します。これにより、ネットワーク内のルートブリッジノードを決定する IEEE STP の優先度の値が設定されます。デフォルト値は 32768 です。有効な値は 0 (もっとも高い優先度) - 61440 (もっとも低い優先度) で、4096 ずつ増分します。

4096 で割り切れない値が使用された場合、システムはその値を自動的に 4096 で割り切れる次の小さい値に切り捨てます。

-m *max-age*, --max-age=*max-age*
構成情報の最長有効期間 (秒単位) を指定します。これにより、STP Bridge Max Age パラメータが設定されます。このノードがルートブリッジである場合は、ネットワーク内のすべてのノードに対してこの値が使用されます。この

時間より前のブリッジリンク情報は破棄されます。デフォルトは 20 秒です。有効な値は 6 - 40 秒です。その他の制約については、 `-d` *forward-delay* パラメータを参照してください。

`-h` *hello-time*, `--hello-time=`*hello-time*
STP Bridge Hello Time パラメータを指定します。このノードがルートノードである場合、このノードは、ネットワーク全体にわたってこの間隔で構成 BPDU を送信します。デフォルト値は 2 秒です。有効な値は 1 - 10 秒です。その他の制約については、 `-d` *forward-delay* パラメータを参照してください。

`-d` *forward-delay*, `--forward-delay=`*forward-delay*
STP Bridge Forward Delay パラメータを指定します。このノードがルートノードである場合、ポートが有効になっていると、ネットワーク内のすべてのブリッジがこのタイマーを使用してリンク状態を順序付けます。デフォルト値は 15 秒です。有効な値は 4 - 30 秒です。

ブリッジは次の 2 つの制約に従う必要があります。

```
2 * (forward-delay - 1.0) >= max-age

max-age >= 2 * (hello-time + 1.0)
```

これらの制約に違反するパラメータ設定はすべてエラーとして扱われるため、コマンドが失敗して診断メッセージが生成されます。このメッセージには、指定された値に代わる有効な値が表示されます。

`-f` *force-protocol*, `--force-protocol=`*force-protocol*
MSTP で強制的にサポートされる最大プロトコルを指定します。デフォルトの値は 3 です。有効な値は、負ではない整数です。現在の実装では RSTP や MSTP がサポートされていないため、現在この値は無視されます。ただし、将来 MSTP が使用されないようにするために、このパラメータを STP のみの場合は `0` に、STP と RSTP の場合は `2` に設定できます。

`-l` *link*, `--link=`*link*
新しく作成されたブリッジに追加する 1 つ以上のリンクを指定します。これは、`add-bridge` サブコマンドのように、ブリッジを作成してから 1 つ以上のリンクを追加する場合と同様です。ただし、いずれかのリンクを追加できない場合は、コマンド全体が失敗し、新しいブリッジ自体が作成されません。同じコマンド行で複数のリンクを追加するには、リンクごとにこのオプションを繰り返します。リンクなしのブリッジも作成できます。リンクの割り当ての詳細については、`add-bridge` サブコマンドを参照してください。

ブリッジの作成およびリンクの割り当てには、`PRIV_SYS_DL_CONFIG` 特権が必要です。オプションのブリッジ機能がシステムにインストールされていない場合、ブリッジの作成が失敗する可能性があります。

`dladm modify-bridge` [`-P` *protect*] [`-R` *root-dir*] [`-p` *priority*] [`-m` *max-age*] [`-h` *hello-time*] [`-d` *forward-delay*] [`-f` *force-protocol*] [`-l` *link*...] *bridge-name*

既存のブリッジの動作パラメータを変更します。これらのオプションは `create-bridge` サブコマンドの場合と同じですが、`-l` オプションが許可されていない点が異なります。既存のブリッジにリンクを追加するには、`add-bridge` サブコマンドを使用します。

ブリッジのパラメータの変更には、`PRIV_SYS_DL_CONFIG` 特権が必要です。

`dladm delete-bridge` [`-R` *root-dir*] *bridge-name*

ブリッジインスタンスを削除します。削除されるブリッジにリンクが追加されていてはいけません。ブリッジを削除する前にリンクを無効にするには、`remove-bridge` サブコマンドを使用します。

ブリッジの削除には、`PRIV_SYS_DL_CONFIG` 特権が必要です。

`-R` (`--root-dir`) オプションは、`create-bridge` サブコマンドの場合と同じです。

`dladm add-bridge` [`-R` *root-dir*] `-l` *link* [`-l` *link*...] *bridge-name*

既存のブリッジに 1 つ以上のリンクを追加します。複数のリンクが指定され、そのいずれかを追加するときにエラーが発生した場合、そのコマンドは失敗し、システムには変更が加えられません。

ブリッジへのリンクの追加には、`PRIV_SYS_DL_CONFIG` 特権が必要です。

1 つのリンクは、最大 1 つのブリッジのメンバーになることができます。すでに別のブリッジに属しているリンクを追加しようとすると、エラーが発生します。あるブリッジインスタンスから別のブリッジインスタンスにリンクを移動するには、現在のブリッジからそのリンクを削除してから、新しいブリッジに追加します。

ブリッジに割り当てるリンクが、同時に VLAN、VNIC、またはトンネルであってはいけません。ブリッジに割り当てられるのは、物理 Ethernet データリンク、集約データリンク、および Ethernet スタブだけです。

ブリッジに割り当てるリンクはすべて、MTU が同じである必要があります。これは、リンクが割り当てられるときにチェックされます。リンクがブリッジ上の最初のリンクでなく、MTU が異なっている場合、そのリンクは無効な形式でブリッジに追加されます。

ブリッジを使用しているシステムでは、`eeprom(1M)` `local-mac-address?` 変数を false に設定するべきではないことに注意してください。

これらのオプションは、`create-bridge` サブコマンドの場合と同じです。

`dladm remove-bridge` [`-R` *root-dir*] `-l` *link* [`-l` *link*...] *bridge-name*

ブリッジインスタンスから 1 つ以上のリンクを削除します。複数のリンクが指定されていて、そのいずれかを削除するときにエラーが発生した場合、そのコマンドは失敗し、何も削除されません。

ブリッジからのリンクの削除には、PRIV_SYS_DL_CONFIG 特権が必要です。

これらのオプションは、create-bridge サブコマンドの場合と同じです。

dladm show-bridge [-flt] [-s [-i *interval*]] [[-p] -o *field*,...] [*bridge-name*]
ブリッジの実行ステータスと構成、追加されているリンク、学習された転送エントリ、および TRILL ニックネームデータベースを表示します。全体的なブリッジのステータスと構成を表示する場合は、ブリッジ名を省略すると、すべてのブリッジを表示できます。その他の形式の場合は、ブリッジの指定が必要です。

show-bridge サブコマンドで使用できるオプションは、次のとおりです。

-i *interval*, --interval=*interval*
統計情報が表示される間隔 (秒) を指定するために -s オプションとともに使用されます。このオプションが指定されていない場合、統計情報は 1 回だけ表示されます。

-s, --statistics
指定されたブリッジ、または特定のブリッジに追加されたリンクの統計情報を表示します。このオプションは -f オプションおよび -t オプションとともに使用することはできません。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。下記の「解析可能な出力形式」を参照してください。

-o *field*[,...], --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名については、あとで説明します。特殊な値 all はすべてのフィールドを表示します。-o が指定されていない場合、フィールドの各セットには、表示する独自のデフォルトセットがあります。

デフォルトでは、show-bridge サブコマンドはブリッジ構成を表示します。次のフィールドを表示できます。

BRIDGE
ブリッジの名前。

ADDRESS
ブリッジの一意の識別子の値 (MAC アドレス)。

PRIORITY
create-bridge および modify-bridge の -p で設定された、構成済みの優先度の値。

BMAXAGE
create-bridge および modify-bridge の -m で設定された、構成済みのブリッジ最長有効期間。

BHELLOTIME
: create-bridge および modify-bridge の -h で設定された、構成済みのブリッジ Hello 時間。

BFWDDELAY
: create-bridge および modify-bridge の -d で設定された、構成済みの転送遅延。

FORCEPROTO
: create-bridge および modify-bridge の -f で設定された、構成済みの強制適用最大プロトコル。

TCTIME
: 最後のトポロジ変更からの時間 (秒単位)。

TCCOUNT
: トポロジ変更の回数。

TCHANGE
: これは、トポロジ変更が検出されたことを示します。

DESROOT
: ルートノードのブリッジ識別子。

ROOTCOST
: ルートノードへのパスのコスト。

ROOTPORT
: ルートノードに到達するために使用されるポート番号。

MAXAGE
: ルートノードからの最長有効期間の値。

HELLOTIME
: ルートノードからの Hello 時間の値。

FWDDELAY
: ルートノードからの転送遅延の値。

HOLDTIME
: 最小の BPDU 間隔。

デフォルトでは、-o オプションが指定されていない場合、BRIDGE、ADDRESS、PRIORITY、および DESROOT フィールドのみが表示されます。

-s オプションが指定されている場合、show-bridge サブコマンドはブリッジの統計情報を表示します。次のフィールドを表示できます。

BRIDGE
: ブリッジ名。

DROPS
リソースの問題のためにドロップされたパケットの数。

FORWARDS
あるリンクから別のリンクに転送されたパケットの数。

MBCAST
ブリッジによって処理されたマルチキャストパケットとブロードキャストパケットの数。

RECV
追加されているすべてのリンク上で受信されたパケットの数。

SENT
追加されているすべてのリンク上で送信されたパケットの数。

UNKNOWN
不明な宛先を含んだ処理済みパケットの数。これらのパケットはすべてのリンクに送信されます。

デフォルトでは、-o オプションが指定されていない場合、BRIDGE、DROPS、および FORWARDS フィールドのみが表示されます。

show-bridge サブコマンドでは、次のオプションも使用できます。

-l, --link
1 つのブリッジインスタンスに追加されているすべてのリンクについて、リンク関連のステータスと統計情報を表示します。-s オプションを指定せずにこのオプションを使用すると、リンクごとに次のフィールドを表示できます。

LINK
リンク名。

INDEX
ブリッジ上のポート (リンク) のインデックス番号。

STATE
リンクの状態。この状態は、disabled、discarding、learning、forwarding、non-stp、bad-mtu のいずれかになります。

UPTIME
最後のリセットまたは初期化からの秒数。

OPERCOST
使用されている実際のコスト (1 - 65535)。

OPERP2P
これは、ポイントツーポイント (P2P) モードが検出されたかどうかを示します。

OPEREDGE
　これは、エッジモードが検出されたかどうかを示します。

DESROOT
　このポート上で確認されたルートブリッジ識別子。

DESCOST
　指定されたポートを経由するネットワークルートノードへのパスコスト。

DESBRIDGE
　このポートのブリッジ識別子。

DESPORT
　このポートの構成メッセージを送信するために使用されるポートの ID と優先度。

TCACK
　これは、トポロジ変更確認応答が検出されたかどうかを示します。

-o オプションなしで -l オプションが指定されている場合、LINK、STATE、UPTIME、および DESROOT フィールドのみが表示されます。

-l オプションが指定されている場合、-s オプションを使用すると、リンクごとに次のフィールドを表示できます。

LINK
　リンク名。

CFGBPDU
　受信された構成 BPDU の数。

TCNBPDU
　受信されたトポロジ変更 BPDU の数。

RSTPBPDU
　受信された Rapid Spanning Tree BPDU の数。

TXBPDU
　送信された BPDU の数。

DROPS
　リソースの問題のためにドロップされたパケットの数。

RECV
　ブリッジによって受信されたパケットの数。

XMIT
　ブリッジによって送信されたパケットの数。

-o オプションが指定されていない場合、LINK、DROPS、RECV、および XMIT フィールドのみが表示されます。

`-f, --forwarding`
1 つのブリッジインスタンスの転送エントリを表示します。このオプションを使用すると、転送エントリごとに次のフィールドを表示できます。

`DEST`
着信先 MAC アドレス。

`AGE`
エントリの有効期間 (秒およびミリ秒単位)。ローカルエントリでは省略されます。

`FLAGS`
MAC アドレスが、追加されたリンクか、またはいずれかの追加されたリンク上の VNIC に属している場合は、`L` (ローカル) フラグが表示されます。

`OUTPUT`
ローカルエントリの場合、これは MAC アドレスを含む追加されたリンクの名前です。それ以外の場合、Spanning Tree Protocol を使用するブリッジでは、これは出力インタフェースの名前になります。RBridge では、これは出力 `TRILL` ニックネームになります。

`-o` オプションが指定されていない場合、`DEST`、`AGE`、`FLAGS`、および `OUTPUT` フィールドが表示されます。

`-t, --trill`
1 つのブリッジインスタンスの `TRILL` ニックネームエントリを表示します。このオプションを使用すると、`TRILL` ニックネームエントリごとに次のフィールドを表示できます。

`NICK`
この RBridge の `TRILL` ニックネーム。これは、1 - 65535 の数字です。

`FLAGS`
このニックネームでローカルシステムが識別される場合は、`L` フラグが表示されます。

`LINK`
この RBridge にメッセージを送信するときの出力のリンク名。

`NEXTHOP`
このニックネームを持つ RBridge に到達するために使用される次のホップの RBridge の MAC アドレス。

`-o` オプションが指定されていない場合、`NICK`、`FLAGS`、`LINK`、および `NEXTHOP` フィールドが表示されます。

`dladm create-vlan` [`-ft`] [`-R` *root-dir*] `-l` *ether-link* `-v` *vid* [*vlan-link*]

Ethernet リンク *ether-link* 上に *vid* の ID を持つタグ付き VLAN リンクを作成します。VLAN リンクの名前は、*vlan-link* として指定できます。名前が指定されていない場合、次の名前が自動的に生成されます (*ether-link* が *namePPA* であると仮定)。

`<`*name*`><1000 * `*vlan-tag*` + `*PPA*`>`

たとえば、*ether-link* が `bge1` で *vid* が 2 の場合、生成される名前は `bge2001` になります。

`-f`, `--force`
: VLAN リンクの作成を強制します。一部のデバイスでは、VLAN ヘッダーを含めることができる大きさのフレームサイズが許可されません。このようなデバイス上に VLAN リンクを作成するには `-f` オプションが必要で、結果として得られる VLAN 上の IP インタフェースの MTU を 1500 ではなく 1496 に設定する必要があります。

`-l` *ether-link*
: VLAN が作成される Ethernet リンクを指定します。

`-t`, `--temporary`
: VLAN リンクを一時的なものとして指定します。一時的な VLAN リンクは、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
: 前述の「オプション」を参照してください。

`dladm delete-vlan` [`-t`] [`-R` *root-dir*] *vlan-link*

指定された VLAN リンクを削除します。

`delete-vlan` サブコマンドで使用できるオプションは、次のとおりです。

`-t`, `--temporary`
: 削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
: 前述の「オプション」を参照してください。

`dladm show-vlan` [`-PZ`] [[`-p`] `-o` *field*[,...]] [`-z` *zone*[,...]] [*vlan-link*]

すべての VLAN リンクまたは指定された VLAN リンクの VLAN 構成を表示します。

`show-vlan` サブコマンドで使用できるオプションは、次のとおりです。

`-o` *field*[,...]、 `--output=`*field*[,...]
: 表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 `all` にする必要があります。VLAN リンクごとに、次のフィールドを表示できます。

LINK
VLAN リンクの名前。

VID
VLAN に関連付けられた ID。

OVER
この VLAN が構成されている物理リンクの名前。

FLAGS
VLAN リンクに関連付けられたフラグのセット。指定可能なフラグは次のとおりです。

f
この VLAN は、create-vlan の -f オプションを使用して作成されました。

i
この VLAN は、DLPI リンクが開かれたときに暗黙的に作成されました。これらの VLAN リンクは、その DLPI リンクが最後に閉じられたときに (たとえば、その VLAN リンクに関連付けられている IP インタフェースが unplumb されたときに) 自動的に削除されます。

将来、追加のフラグが定義される可能性があります。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
実行中のシステムの状態ではなく、永続的な VLAN 構成を表示します。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm scan-wifi [[-p] -o *field*[,...]] [*wifi-link*]
すべての WiFi リンク上、または指定された *wifi-link* 上のみにある WiFi ネットワークを調べます。

デフォルトでは、BSSTYPE を除く現在のすべてのフィールドが表示されます。

-o *field*[,...], --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。見つかった WiFi ネットワークごとに、次のフィールドを表示できます。

LINK
WiFi ネットワークが存在するリンクの名前。

ESSID
WiFi ネットワークの ESSID (名前)。

BSSID
WiFi ネットワークのアクセスポイントのハードウェアアドレス (BSS ネットワークの場合)、または WiFi ネットワークのランダムに生成された一意のトークン (IBSS ネットワークの場合) のどちらか。

SEC
none (セキュリティーを使用しない WiFi ネットワークの場合)、wep (WEP (Wired Equivalent Privacy) が必要な WiFi ネットワークの場合)、または wpa (WPA (Wi-Fi Protected Access) が必要な WiFi ネットワークの場合) のいずれか。

MODE
サポートされている接続モード (a、b、g、n のうちの 1 つまたは複数)。

STRENGTH
信号の強さ (excellent、very good、good、weak、very weak のうちのいずれか)。

SPEED
WiFi ネットワークの最大速度 (Mbps 単位)。

BSSTYPE
bss (BSS (インフラストラクチャー) ネットワークの場合)、または ibss (IBSS (アドホック) ネットワークの場合) のどちらか。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

dladm connect-wifi [-e *essid*] [-i *bssid*] [-k *key*,...] [-s none | wep | wpa] [-a open|shared] [-b bss|ibss] [-c] [-m a|b|g|n] [-T *time*] [*wifi-link*]

WiFi ネットワークに接続します。これは、検出、フィルタ処理、優先順位付け、および関連付けの 4 つの手順で構成されます。ただし、非ブロードキャスト WiFi ネットワークへの接続を可能にし、パフォーマンスを向上させるために、-e または -i オプションを使用して BSSID または ESSID が指定されている場合、最初の 3 つの手順は省略され、connect-wifi は、指定された残りのパラメータに一致する BSSID または ESSID への関連付けをただちに試みます。この関連付けが失敗したものの、指定された条件に一致するほかのネットワークが存在する可能性がある場合は、後述する従来の発見プロセスが開始されます。

発見の手順では、指定された WiFi リンク上の使用可能なすべての WiFi ネットワークが検索されますが、この段階では接続を実行してはいけません。管理の便宜上、システム上に WiFi リンクが 1 つしか存在しない場合は、*wifi-link* を省略できます。

検出が完了すると、次のオプションの値に従ってネットワークのリストがフィルタ処理されます。

-e *essid*, --essid=*essid*
同じ *essid* を持たないネットワークがフィルタ処理によって除外されます。

-b bss|ibss, --bsstype=bss|ibss
同じ bsstype を持たないネットワークがフィルタ処理によって除外されます。

-m a|b|g, --mode=a|b|g|n
指定された 802.11 モードに適していないネットワークがフィルタ処理によって除外されます。

-k *key,...*, --key=*key, ...*
キーで指定された secobj を使用してネットワークに接続します。指定されたキーに適していないネットワークがフィルタ処理によって除外されます。

-s none|wep|wpa, --sec=none|wep|wpa
指定されたセキュリティーモードに適していないネットワークがフィルタ処理によって除外されます。

次に、残りのネットワークが、最初に信号強度で、次に最大速度で優先付けされます。最後に、リスト内の各ネットワークに (順番に) 関連付けする試みが、1 つのネットワークで成功するか、または残りのネットワークがなくなるまで行われます。

前述のオプションに加えて、次のオプションも connect-wifi の動作を制御します。

-a open|shared, --auth=open|shared
指定された認証モードを使用して接続します。デフォルトでは、open と shared が順番に試行されます。

-c, --create-ibss
指定された ESSID に一致するアドホックネットワークが見つからない場合に、新しいアドホックネットワークを作成するために -b ibss とともに使用されます。ESSID が指定されていない場合、 -c -b ibss は、常に新しいアドホックネットワークの作成を開始します。

-T *time*, --timeout=*time*
関連付けが成功するまで待機する秒数を指定します。*time* を forever にすると、関連付けを無期限に待機します。現在のデフォルト値は 10 秒ですが、将来変更される可能性があります。デフォルト値より短いタイムアウトは、成功しない可能性があります。

-k *key,...*, --key=*key,...*
前述のフィルタ処理に加えて、関連付けが確実に行われるようにするために、指定されたキーが使用されます。使用するセキュリティーモードはキークラスに基づきます。セキュリティーモードが明示的に指定された場

合、そのセキュリティーモードはキークラスと互換性を持っている必要があります。すべてのキーが同じクラスに属している必要があります。

複数のキースロットをサポートするセキュリティーモードの場合、キーを格納するスロットは、コロンとそれに続くインデックスで指定されます。したがって、`-k mykey:3` と指定すると、`mykey` がスロット 3 に格納されます。デフォルトでは、スロット 1 とみなされます。複数のキーをサポートするセキュリティーモードの場合、最初のキーをアクティブなキーにしたコンマ区切りリストを指定できます。

`dladm disconnect-wifi` [`-a`] [*wifi-link*]

1 つ以上の WiFi ネットワークとの接続を切断します。接続済みの WiFi リンクが *wifi-link* で指定されている場合は、そのリンクが切断されます。管理の便宜上、WiFi リンクが 1 つしか接続されていない場合は、*wifi-link* を省略できます。

`-a`, `--all-links`

すべての接続されているリンクを切断します。これは、スクリプトによる使用を主に想定しています。

`dladm show-wifi` [`-Z`] [[`-p`] `-o` *field*,...] [`-z` *zone*[,...]] [*wifi-link*]

すべての WiFi リンクまたは指定されたリンク *wifi-link* の WiFi 構成情報を表示します。

`-o` *field*,..., `--output=`*field*

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 `all` にする必要があります。WiFi リンクごとに、次のフィールドを表示できます。

`LINK`

表示されているリンクの名前。

`STATUS`

`connected` (リンクが接続されている場合)、または `disconnected` (リンクが接続されていない場合) のどちらか。リンクが切断されている場合は、残りのすべてのフィールドに値 `--` が表示されます。

`ESSID`

接続されている WiFi ネットワークの ESSID (名前)。

`BSSID`

WiFi ネットワークのアクセスポイントのハードウェアアドレス (BSS ネットワークの場合)、または WiFi ネットワークのランダムに生成された一意のトークン (IBSS ネットワークの場合) のどちらか。

`SEC`

`none` (セキュリティーを使用しない WiFi ネットワークの場合)、`wep` (WEP が必要な WiFi ネットワークの場合)、または `wpa` (WPA が必要な WiFi ネットワークの場合) のいずれか。

MODE
サポートされている接続モード (a、b、g、n のうちの 1 つまたは複数)。

STRENGTH
接続の強さ (excellent、very good、good、weak、very weak のうちのいずれか)。

SPEED
接続速度 (Mbps 単位)。

AUTH
open または shared のどちらか (connect-wifi を参照)。

BSSTYPE
bss (BSS (インフラストラクチャー) ネットワークの場合)、または ibss (IBSS (アドホック) ネットワークの場合) のどちらか。

デフォルトでは、AUTH、BSSID、BSSTYPE を除く現在のすべてのフィールドが表示されます。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm show-ether [-xZ] [[-p] -o *field*,...] [-z *zone*[,...]] [*ether-link*]
すべての物理 Ethernet リンクまたは指定された物理 Ethernet リンクの状態情報を表示します。

show-ether サブコマンドで使用できるオプションは、次のとおりです。

-o *field*,..., --output=*field*
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。リンクごとに、次のフィールドを表示できます。

LINK
表示されているリンクの名前。

PTYPE
パラメータのタイプ。ここで、current はリンクのネゴシエーションの状態を、capable はデバイスでサポートされている機能を、adv は通知された機能を、peeradv はリンクパートナーによって通知された機能を示します。

STATE
リンクの状態。

AUTO
自動ネゴシエーションが通知されるかどうかを示す yes/no の値。

SPEED-DUPLEX
指定可能な速度と二重通信の値の組み合わせ。速度の単位は、G (Gbps) または M (Mbps) の接尾辞でエンコードされます。二重通信の値は、f (全二重) または h (半二重) としてエンコードされます。

PAUSE
フロー制御情報。no (フロー制御が使用できないことを示す場合)、tx (エンドポイントは一時停止フレームを送信できるものの、受信した一時停止フレームをすべて無視することを示す場合)、rx (エンドポイントは一時停止フレームを受信し、受信した一時停止フレームに基づいて動作することを示す場合)、bi (双方向のフロー制御を示す場合) のいずれかです。

REM_FAULT
障害検出情報。有効な値は none または fault です。

デフォルトでは、REM_FAULT を除くすべてのフィールドが「現在」の PTYPE で表示されます。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-x, --extended
current、capable、adv、および peeradv の PTYPE 値に対して、拡張出力が表示されます。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm set-linkprop [-t] [-R *root-dir*] -p *prop*=*value*[,...] *link*
指定されたリンク上の 1 つ以上のプロパティーの値を設定します。プロパティーとそれらのプロパティーで指定可能なリストは、リンクタイプ、ネットワークデバイスドライバ、およびネットワーキングハードウェアによって異なります。これらのプロパティーは、show-linkprop を使用すると取得できます。

-t, --temporary
変更を一時的なものとして指定します。一時的な変更は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-p *prop*=*value*[,...], --prop *prop*=*value*[,...]
指定された値に設定するプロパティーのコンマ区切りのリスト。

永続的な値が設定されると、一時的な値が同じ値に変更されることに注意してください。

dladm reset-linkprop [-t] [-R *root-dir*] [-p *prop*,...] *link*
1 つ以上のプロパティーを、指定されたリンク上のそれらの値にリセットします。プロパティーは、起動時に設定されていた値にリセットされます。プロパティーが指定されていない場合、すべてのプロパティーがリセットされます。プロパティーについては、show-linkprop を参照してください。

-t, --temporary
リセットを一時的なものとして指定します。値はデフォルト値にリセットされます。一時的なリセットは、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-p *prop*, ..., --–prop=*prop*, ...
リセットするプロパティーのコンマ区切りのリスト。

永続的な値がリセットされると、一時的な値は同じ値に変更されます。

dladm show-linkprop [-PZ] [[-c] -o *field*[,...]][-p *prop*[,...]] [-z *zone*[,...]] [*link*]
すべてのデータリンクまたは指定されたリンクについて、1 つ以上のプロパティーの現在の値または永続的な値を表示します。デフォルトでは、現在の値が表示されます。プロパティーが指定されていない場合、使用可能なすべてのリンクプロパティーが表示されます。プロパティーごとに、次のフィールドが表示されます。

-o *field*[,...], --output=*field*
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。リンクごとに、次のフィールドを表示できます。

LINK
データリンクの名前。

PROPERTY
プロパティーの名前。

PERM
プロパティーの読み取り権と書き込み権。表示される値は ro または rw のいずれかです。

VALUE
現在の (または永続的な) プロパティー値。値が設定されていない場合は、-- と表示されます。値が不明な場合は、? と表示されます。設定され

ていないか、またはリセットされた永続的な値は -- と表示され、システムの DEFAULT 値 (ある場合) を使用します。

DEFAULT
プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、-- が表示されます。

POSSIBLE
プロパティーに設定できる値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形として *min* - *max* が表示される可能性があります。指定可能な値が不明または無限の場合は、-- が表示されます。

プロパティーのリストはリンクタイプやネットワークデバイスドライバによって異なり、特定のプロパティーに対して指定可能な値はさらに、基本となるネットワークハードウェアやその状態によって異なります。一般的なリンクのプロパティーについては、「一般的なリンクのプロパティー」の節で説明されています。ただし、「_」(下線) で始まるリンクプロパティーは特定のリンクまたはその基礎となるネットワークデバイスに固有であり、変更または削除される可能性があります。詳細については、該当するネットワークデバイスドライバのマニュアルページを参照してください。

-c, --parseable
安定したマシン解析可能な形式を使用して表示します。このオプションには -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
永続的なリンクプロパティーの情報を表示します。

-p *prop*, ..., --prop=*prop*, ...
表示するプロパティーのコンマ区切りのリスト。サブコマンドの説明に続くリンクプロパティーの節を参照してください。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm create-secobj [-t] [-R *root-dir*] [-f *file*] -c *class secobj*
指定された *class* 内に、暗号化されたネットワークへの接続時に WEP または WPA キーとしてあとで使用される *secobj* という名前のセキュリティー保護されたオブジェクトを作成します。セキュリティー保護されたオブジェクトの値は、対話形式で指定することも、ファイルから読み取ることもできます。対話形式のプロンプトの並びやファイル形式は、セキュリティー保護されたオブジェクトのクラスによって異なります。

現在、wep クラスと wpa クラスがサポートされています。WEP (Wired Equivalent Privacy) キーには、5 または 13 バイト長のいずれかを指定できます。ASCII 文字

列または 16 進数文字列として指定できるため、`12345` と `0x3132333435` は同じ 5 バイトキーです (接頭辞 `0x` は省略できます)。WEP キーを含むファイルは、いずれかの WEP キー形式を使用した 1 行で構成されている必要があります。WPA (Wi-Fi Protected Access) キーは、8 - 63 バイトまでの長さを持つ ASCII 文字列として指定する必要があります。

このサブコマンドは、「ネットワークリンクセキュリティー」RBAC プロファイルに属しているユーザーまたは役割のみが使用できます。

`-c` *class*, `--class=`*class*
class は `wep` または `wpa` のいずれかです。前の説明を参照してください。

`-t`, `--temporary`
作成を一時的なものとして指定します。一時的な作成は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`-f` *file*, `--file=`*file*
セキュリティー保護されたオブジェクトの値を取得するために使用するべきファイルを指定します。このファイルの形式は、セキュリティー保護されたオブジェクトのクラスによって異なります。このオプションを使用して WEP キーを設定する例については、「使用例」の節を参照してください。

`dladm delete-secobj` [`-t`] [`-R` *root-dir*] *secobj*[,...]
指定された 1 つ以上のセキュリティー保護されたオブジェクトを削除します。このサブコマンドは、「ネットワークリンクセキュリティー」RBAC プロファイルに属しているユーザーまたは役割のみが使用できます。

`-t`, `--temporary`
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`dladm show-secobj` [`-P`] [[`-p`] `-o` *field*[,...]] [*secobj*,...]
セキュリティー保護された現在のオブジェクトまたは永続オブジェクトの情報を表示します。1 つ以上のセキュリティー保護されたオブジェクトが指定されている場合は、各オブジェクトの情報が表示されます。それ以外の場合は、セキュリティー保護された現在のオブジェクトまたは永続オブジェクトがすべて表示されます。

デフォルトでは、セキュリティー保護された現在のオブジェクトが表示されます。これは、永続的に作成されてから一時的に削除されていないか、または一時的に作成されたすべてのセキュリティー保護されたオブジェクトです。

セキュリティー上の理由から、セキュリティー保護されたオブジェクトの値を表示することはできません。

-o *field*[,...], --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれかである必要があります。表示されるセキュリティー保護されたオブジェクトの場合は、次のフィールドを表示できます。

OBJECT
セキュリティー保護されたオブジェクトの名前。

CLASS
セキュリティー保護されたオブジェクトのクラス。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
セキュリティー保護された永続オブジェクトの情報を表示します。

dladm create-vnic [-t] -l *link* [-R *root-dir*] [-m *value* | auto | {factory [-n *slot-identifier*]} | {vrrp -A {inet | inet6} -V *vrid*} | {random [-r *prefix*]}] [-v *vlan-id*] [-p *prop*=*value* [,...]] *vnic-link*
指定されたリンク上に *vnic-link* という名前の VNIC を作成します。

-t, --temporary
VNIC を一時的なものとして指定します。一時的な VNIC は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-l *link*, --link=*link*
link には物理リンクまたは etherstub を指定できます。

-m *value* | *keyword*, --mac-address=*value* | *keyword*
指定された値またはキーワードに基づいて VNIC の MAC アドレスを設定します。*value* がキーワードでない場合は、ユニキャスト MAC アドレスとして解釈されます。これは、基礎となる NIC にとって有効なアドレスである必要があります。ユーザー指定の MAC アドレスは、MAC アドレスのグローバルで一意のタイプおよびローカルで管理されるタイプによって指定された範囲から取得する必要があります。

次の特殊なキーワードを使用できます。

factory [-n *slot-identifier*],
factory [--slot=*slot-identifier*]

ファクトリ MAC アドレスを VNIC に割り当てます。ファクトリ MAC アドレスが要求される場合は、-m を -n オプションと組み合わせて、MAC アドレススロットを指定して使用できます。-n が指定されていない場合、システムは、次に使用可能なファクトリ MAC アドレスを選択します。show-phys サブコマンドの -m オプションを使用すると、ファクトリ MAC アドレス、そのスロット識別子、およびその可用性を一覧表示できます。

random [-r *prefix*],
random [--mac-prefix=*prefix*]
ランダムな MAC アドレスを VNIC に割り当てます。ローカルビットが設定された有効な IEEE OUI で構成されたデフォルトの接頭辞が使用されます。その接頭辞は、-r オプションを使用すると上書きできます。

vrrp -A {inet | inet6} -V *vrid*
指定されたアドレスファミリと *vrid* に基づいて、VRRP 仮想 MAC アドレスを VNIC に割り当てます。

auto
最初に、ファクトリ MAC アドレスの使用を試みます。どれも使用できない場合は、ランダムな MAC アドレスを割り当てます。-m オプションが指定されていない場合は、auto がデフォルトの動作です。

-v *vlan-id*
この VNIC の VLAN タグ付けを有効にします。VLAN タグには、ID *vlan-id* が割り当てられます。

-p *prop=value,...*, --prop *prop=value,...*
指定された値に設定するプロパティーのコンマ区切りのリスト。

dladm delete-vnic [-t] [-R *root-dir*] *vnic-link*
指定された VNIC を削除します。

-t, --temporary
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

dladm show-vnic [-pPZ] [-s [-i *interval*]] [-o *field*[,...]] [-l *link*] [-z *zone*[,...]] [*vnic-link*]
すべての VNIC、リンク上のすべての VNIC、または指定された *vnic-link* のみの VNIC 構成情報を表示します。

-o *field*[,...] , --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、show-vnic はすべてのフィールドを表示します。

LINK
VNIC の名前。

OVER
この VNIC が構成されている物理リンクの名前。

SPEED
VNIC の最大速度 (Mbps 単位)。

MACADDRESS
VNIC の MAC アドレス。

MACADDRTYPE
VNIC の MAC アドレスタイプ。dladm は、次の MAC アドレスタイプを識別します。

random
VNIC に割り当てられたランダムなアドレス。

factory
VNIC によって使用されるファクトリ MAC アドレス。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-P, --persistent
永続的な VNIC 構成を表示します。

-s, --statistics
VNIC の統計情報を表示します。このオプションは dlstat(1M) によって廃止されています。

-i *interval*, --interval=*interval*
統計情報が表示される間隔 (秒) を指定するために -s オプションとともに使用されます。このオプションは dlstat(1M) によって廃止されています。

-l *link*, --link=*link*
指定されたリンク上のすべての VNIC の情報を表示します。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm create-part [-t] [-f] [-R *root-dir*] -l *ib-link* [-p *prop*=*value*[,..]] - P *pkey part-link*
指定されたリンク上に *part-link* という名前の IP-over-IB リンクを作成します。このサブコマンドは、InfiniBand 物理リンク上でのみサポートされます。

`-f`, `--force`
pkey がポート上に存在しない場合、マルチキャストグループが存在しない場合、またはポートが停止している場合でも、パーティションリンクを強制的に作成します。

`-l` *ib-link*, `--link=`*ib-link*
IP-over-IB 物理リンク名。

`-P`, `--pkey=`*pkey*
パーティションリンクを作成するために使用されるパーティションキー。指定された *pkey* は、接頭辞 `0x` が付いているかどうかにかかわらず、常に 16 進数として扱われます。

`-p` *prop=value*[,..]
`--prop` *prop=value*[,..]
指定された値に設定するプロパティーのコンマ区切りのリスト。サポートされるプロパティーについては、後述の「一般的なリンクのプロパティー」の節で説明します。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`-t`, `--temporary`
パーティションリンクの作成を一時的なものとして指定します。一時的なパーティションリンクは、次回のリブートまで保持されます。

`dladm delete-part` [`-R` *root-dir*] *part-link*
指定されたパーティションリンクを削除します。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`-t`, `--temporary`
パーティションリンクの削除が一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

`dladm show-part` [`-pP`] [`-l` *ib-link*] [`-o` *field*[,...]] [*part-link*]
すべてのパーティションリンク、*ib-link* 上のすべてのパーティション、または指定された *part-link* のみの IB パーティションリンク情報を表示します。

`-l` *ib-link*, `--link=`*ib-link*
指定されたリンク上のすべてのパーティションの情報を表示します。

`-o` *field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 `all` にする必要があります。デフォルトでは (`-o` の指定なし)、`show-part` はすべてのフィールドを表示します。

`LINK`
パーティションリンクの名前。

PKEY
パーティションリンクに関連付けられた Pkey。

OVER
このパーティションリンクが作成されている物理リンクの名前。

STATE
パーティションリンクの現在の状態。指定可能な値は、up、down、unknown のいずれかになります。

FLAGS
パーティションリンクを作成するために使用される状態フラグのセット。指定可能な値は、次のとおりです。

f
パーティションは (パーティションが作成可能かどうかがチェックされることなく) 強制的に作成されました。

t
パーティションリンクは一時的なものであり、次回のリブートまでしか保持されません。

-P, --persistent
永続的な IB パーティションリンク構成を表示します。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

dladm show-ib [-pP] [-o *field*[,...]] [*ib-link*]
すべての IB リンクまたは指定された IB リンク上の IB 物理リンク情報を表示します。

-o *field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、show-ib はすべてのフィールドを表示します。

LINK
物理リンクの名前。

HCAGUID
HCA のグローバル一意識別子。

PORTGUID
ポートのグローバル一意識別子。

PORT
ポート番号。

STATE
物理リンクの現在の状態。指定可能な値は、up、down、unknown のいずれかになります。

PKEYS
LINK フィールドで指定された IP-over-IB リンクに関連付けられたポート上で使用可能な Pkey。

-P, --persistent
永続的な IB 物理リンク構成を表示します。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

dladm create-etherstub [-t] [-R *root-dir*] *etherstub*
指定された名前で etherstub を作成します。

-t, --temporary
etherstub を一時的なものとして指定します。一時的な etherstub は、リブート後には無効になります。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

VNIC は、物理 NIC ではなく etherstub 上に作成できます。物理 NIC と同様に、このような作成によって、スタックは、同じ etherstub 上に作成された VNIC 間に仮想スイッチを暗黙的に作成します。

dladm delete-etherstub [-t] [-R *root-dir*] *etherstub*
指定された etherstub を削除します。

-t, --temporary
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

dladm show-etherstub [-Z] [-z *zone*[,...]] [*etherstub*]
デフォルトでは、構成されているすべての etherstub を表示します。*etherstub* が指定されている場合は、指定された etherstub を表示します。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm create-iptun [-t] [-R *root-dir*] -T *type* [-a {local|remote}=*addr*,...] *iptun-link*
iptun-link という名前の IP トンネルリンクを作成します。ipsecconf(1M) を使用すると、このようなリンクを IPsec でさらに保護できます。

IP トンネルは、概念的に 2 つの部分で構成されます。1 つは、2 つ以上の IP ノード間の仮想リンク、もう 1 つはこのリンク上の IP インタフェースです。このインタフェースを使用すると、基礎となるリンクによってカプセル化された IP パケットをシステムで送受信できます。このサブコマンドは、仮想リンクを作成します。そのリンク上の IP インタフェースを構成するには、`ipadm(1M)` コマンドを使用します。

`-t`, `--temporary`
IP トンネルリンクを一時的なものとして指定します。一時的なトンネルは、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*
前述の「オプション」を参照してください。

`-T` *type*, `--tunnel-type=`*type*
作成されるトンネルのタイプを指定します。タイプは、次のいずれかである必要があります。

`ipv4`
2 つの IPv4 ノード間のポイントツーポイントな IP-over-IP トンネル。このタイプのトンネルを機能させるには、IPv4 発信元および着信先アドレスが必要です。このようなトンネル上で IPv4 インタフェースと IPv6 インタフェースを plumb すると、IPv4-over-IPv4 および IPv6-over-IPv4 トンネル構成を作成できます。

`ipv6`
IETF RFC 2473 で定義されている、2 つの IPv6 ノード間のポイントツーポイントな IP-over-IP トンネル。このタイプのトンネルを機能させるには、IPv6 発信元および着信先アドレスが必要です。このようなトンネルの上で IPv4 インタフェースと IPv6 インタフェースを plumb すると、IPv4-over-IPv6 および IPv6-over-IPv6 トンネル構成を作成できます。

`6to4`
IETF RFC 3056 で定義されている、6to4 なポイントツーマルチポイントトンネル。このタイプのトンネルを機能させるには、IPv4 発信元アドレスが必要です。このようなトンネルリンク上で IPv6 インタフェースを plumb すると、6to4 ルーターを構成できます。

`-a` {`local`|`remote`}=*addr*,...
`--address` {`local`|`remote`}=*addr*,...
ローカルまたはリモートのトンネルアドレスに対応するリテラル IP アドレスまたはホスト名。ローカルまたはリモートのどちらかを個別に指定することも、その両方をコンマで区切って指定することもできます (たとえば、`-a local=`*laddr*`,remote=`*raddr*)。

`dladm modify-iptun` [`-t`] [`-R` *root-dir*] `-a` {`local`|`remote`}=*addr*,... *iptun-link*
指定された IP トンネルのパラメータを変更します。

-t, --temporary
変更を一時的なものとして指定します。一時的な変更は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

-a {local|remote}=*addr*,...
--address {local|remote}=*addr*,...
トンネルリンクの新しいローカルアドレスまたはリモートアドレスを指定します。説明については、create-iptun を参照してください。

dladm delete-iptun [-t] [-R *root-dir*] *iptun-link*
指定された IP トンネルリンクを削除します。

-t, --temporary
削除を一時的なものとして指定します。一時的な削除は、次回のリブートまで保持されます。

-R *root-dir*, --root-dir=*root-dir*
前述の「オプション」を参照してください。

dladm show-iptun [-PZ] [[-p] -o *field*[,...]] [-z *zone*[,...]] [*iptun-link*]
1 つの IP トンネルまたはすべての IP トンネルの IP トンネルリンク構成を表示します。

-P, --persistent
永続的な IP トンネル構成を表示します。

-p, --parseable
安定したマシン解析可能な形式を使用して表示します。-p には -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-o *field*[,...] , --output=*field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、show-iptun はすべてのフィールドを表示します。

LINK
IP トンネルリンクの名前。

TYPE
create-iptun の -T オプションで指定されたトンネルのタイプ。

FLAGS
IP トンネルリンクに関連付けられたフラグのセット。指定可能なフラグは次のとおりです。

s
IP トンネルリンクは IPsec ポリシーによって保護されます。トンネルリンクに関連付けられた IPsec ポリシーを表示するには、次のように入力します。

```
# ipsecconf -ln -i tunnel-link
```

IPsec ポリシーを構成する方法の詳細については、ipsecconf(1M) を参照してください。

i
この IP トンネルリンクは ipadm(1M) を使用して暗黙的に作成されたもので、参照されなくなると (つまり、そのトンネルを経由する最後の IP インタフェースが削除されると) 自動的に削除されます。暗黙的なトンネル作成の詳細については、ipadm(1M) を参照してください。

LOCAL
ローカルのトンネルアドレス。

REMOTE
リモートのトンネルアドレス。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
上記の dladm show-link の下にある -z オプションの説明を参照してください。

dladm show-usage [-a] -f *filename* [-p *plotfile* -F *format*] [-s *time*] [-e *time*] [*link*]
このサブコマンドは dlstat(1M) show-link -h コマンドによって廃止されています。

解析可能な出力形式

dladm サブコマンドの多くには、出力をマシン解析可能な形式で表示するオプションがあります。出力形式は、コロン (:) で区切られたフィールドで構成された 1 つ以上の行です。表示されるフィールドは、使用されているサブコマンドに固有のもので、特定のサブコマンドの -o オプションに対するエントリの下に一覧表示されます。出力には、-o オプションを使用して要求されたフィールドのみが、要求された順序で表示されます。

複数のフィールドを要求した場合、すべてのコロン文字は、バックスラッシュ (\) でエスケープされてから出力されます。同様に、バックスラッシュ文字もエスケープされます (\\)。シェルの read(1) 関数を環境変数 IFS=: とともに使用すると、このエスケープ形式を解析できます (下の「使用例」を参照)。1 つのフィールドのみを要求した場合はエスケープが実行されないことに注意してください。

一般的なリンクのプロパティー

次の一般的なリンクのプロパティーがサポートされています。

autopush
対応する DLPI デバイスが開かれたときに、あるリンクに関連付けられたストリームにプッシュする STREAMS モジュールのセットを指定します。これは、モジュールのスペース区切りリストです。

オプションの特殊文字シーケンス [anchor] は、ストリーム上にある、リストで以前に指定されたモジュールに STREAMS アンカーを配置するよう指示します。複数のアンカーを指定したり、リストの先頭にアンカーを配置したりするとエラーになります。

autopush プロパティーは、より汎用的な autopush(1M) コマンドよりも優先されます。

cpus
特定のデータリンクのパケットの処理を 1 つのプロセッサまたはプロセッサのセットにバインドします。この値は、1 つ以上のプロセッサ ID のコンマ区切りリストか、または ID の範囲として指定できます。リストが複数のプロセッサで構成されている場合は、処理がすべてのプロセッサに分散されます。各接続について、プロセッサアフィニティーへの接続とパケットの順序は保持されます。

プロセッサまたはプロセッサのセットは、リンク用に排他的に予約されます。そのリンクの処理に関連付けられたカーネルスレッドと割り込みのみが、指定されたプロセッサまたはプロセッサのセットにバインドされます。指定のプロセッサをそのリンク専用にする場合は、psrset(1M) を使用すると、プロセッサセットを作成し、そのプロセッサセットからリンクをバインドするプロセッサを指定できます。

以前の操作によってリンクがすでにプロセッサまたはプロセッサのセットにバインドされている場合、そのバインドは削除され、代わりに新しいプロセッサセットが使用されます。

デフォルトでは、CPU はバインドされません。つまり、パケットの処理は、どの特定のプロセッサまたはプロセッサセットにもバインドされません。

pool リンクプロパティーが設定されているリンクでは、cpus プロパティーを指定できません。

cpus-effective
この読み取り専用プロパティーは、指定されたデータリンク上のパケット処理に使用される CPU のリストを表示します。

cpus プロパティーが設定されている場合、cpus-effective は同じ設定になります。

pool プロパティーが設定されている場合、cpus-effective は、管理者によって指定されたプールから選択されます。

pool プロパティーも cpus プロパティーも設定されていない場合、システムは cpus-effective の適切な値を選択します。

rxfanout
受信側のファンアウトスレッドの数を指定できるようにします。

受信リング上で受信されるトラフィックを複数のスレッドにファンアウトし、並行して処理することができます。これは特に、システムに多数の CPU が搭載されている場合に有効です。このプロパティーは、特定のデータリンクの受信側のファンアウトスレッドの数を表すカウントです。このプロパティーによって管理者は必要な rxfanout を指定できます。ただし、利用可能な CPU とハードウェア RX リングの数に基づいて、システムは異なる (より小さいか、さらに大きい) ファンアウト値を選択する場合があります。

rxfanout-effective
: CPU の数が受信側のファンアウトの上限となり、rxrings の数が下限となります。このため、受信側の実際のファンアウト数は、ユーザーが設定したものと異なる可能性があります。

learn_limit
: ブリッジリンク上で学習される新しい MAC 発信元または変更された MAC 発信元の数を制限します。その数がこの値を超えると、そのリンク上での学習は一時的に無効になります。このプロパティーは、VLAN および VNIC 以外のタイプのリンクにのみ存在します。

  デフォルト値は 1000 です。有効な値は 0 以上です。

learn_decay
: learn_limit によって制限された発信元の変更の減衰率を指定します。この数値は、5 秒ごとにブリッジリンクのカウンタから減算されます。このプロパティーは、VLAN および VNIC 以外のタイプのリンクにのみ存在します。

  デフォルト値は 200 です。有効な値は 0 以上です。

mac-address
: データリンクのプライマリ MAC アドレスを設定します。設定されている場合は、基本となるデータリンクの現在および将来のすべての MAC クライアントによって使用されるプライマリ MAC アドレスを変更します。

maxbw
: リンクの全二重帯域幅を設定します。この帯域幅は、いずれかのスケールサフィックス (Kbps、Mbps、または Gbps を示す K、M、または G) が付いた整数として指定されます。単位が指定されていない場合、入力値は Mbps として読み取られます。デフォルトでは、帯域幅の制限はありません。

pool
: 特定のデータリンクのパケットの処理を、poolcfg(1M) と pooladm(1M) によって定義および管理されるプロセッサのプールにバインドします。この処理のバインドは、CPU のリストが明示されず、代わりにプール機能によって保持される点を除き、cpus リンクプロパティーで実行される処理と同様です。

  プールが有効になっており、リンクに対してプールが指定されていない場合は、pool_default がパケット処理に使用されます。

ip-type=exclusive が設定されているゾーンで、あるプールが pool zone プロパティーまたは dedicated-cpus 割り当てによって指定されている場合は、そのプールもゾーンに関連付けられているすべてのデータリンクに使用されます。

cpus リンクプロパティーが設定されているリンクでは pool プロパティーを指定できません。

pool-effective
プール機能が有効になっている場合、この読み取り専用プロパティーは、パケット処理に使用されているプールを表示します。管理者がデータリンクにプールを割り当てていない場合、プールは pool_default になります。

プール機能が無効になっている場合、有効なプールは存在せず、この値は空になります。

priority
リンクの相対的な優先度を設定します。この値は、トークン high、medium、low のいずれかとして指定できます。デフォルト値は high です。この優先順位は、回線上のどのプロトコル優先順位フィールドでも反映されませんが、システム内のパケット処理スケジューリングに使用されます。

rxringsavail
受信側で使用可能なリングの数を指定する読み取り専用プロパティー。

rxrings
MAC クライアントの受信リングの数を指定します。sw という値は、この MAC クライアントに RX リングが割り当てられず、ソフトウェアベースになることを意味します。hw という値は、この MAC クライアントが RX リング (使用可能な場合) を 1 つ取得でき、そうでなければ、ソフトウェアベースになることを意味します。ゼロ以外の値は、この MAC クライアントに対してその数だけリングを予約し (使用可能な場合)、そうでない場合は失敗することを意味します。このプロパティーを指定しない場合、MAC クライアントは RX リング (使用可能な場合) を 1 つ取得でき、そうでなければソフトウェアベースになります。

rxhwclntavail
作成可能な追加の RX ハードウェアベースの MAC クライアントの数を指定する読み取り専用プロパティー。

txringsavail
送信側で使用可能なリングの数を指定する読み取り専用プロパティー。

txrings
MAC クライアントの送信リングの数を指定します。sw という値は、この MAC クライアントに TX リングが割り当てられないことを意味します。hw という値は、この MAC クライアントが TX リング (使用可能な場合) を 1 つ取得でき、そうでなければ、ソフトウェアベースになることを意味します。ゼロ以外の値は、この MAC クライアントに対してその数だけリングを予約し (使用可能な場合)、そうでない場合は失敗することを意味します。このプロパティーを指定し

ない場合、MAC クライアントは TX リング (使用可能な場合) を 1 つ取得でき、そうでなければソフトウェアベースになります。

`txhwclntavail`
作成可能な追加の TX ハードウェアベースの MAC クライアントの数を指定する読み取り専用プロパティー。

`stp`
ブリッジリンク上の Spanning Tree Protocol を有効または無効にします。この値を `0` に設定すると、スパニングツリーが無効になり、リンクが BPDU 保護の有効な転送モードになります。このモードは、エンドノードにのみ接続されたポイントツーポイントリンクに適しています。このプロパティーは、VLAN および VNIC 以外のタイプのリンクにのみ存在します。デフォルト値は `1` で、STP が有効になります。

`forward`
VLAN の転送を有効または無効にします。この値を `0` に設定すると、VLAN リンクのブリッジ転送が無効になります。ブリッジ転送を無効にすると、その VLAN がブリッジの「許可されたセット」から削除されます。デフォルト値は `1` で、構成された VLAN のブリッジ転送が有効になります。

`default_tag`
デフォルトの VLAN ID を、このリンクとの間で送受信されるタグなしパケットとして設定します。このプロパティーは、VLAN および VNIC 以外のタイプのリンクにのみ存在します。この値を `0` に設定すると、ポートとの間で送受信されるタグなしパケットのブリッジ転送が無効になります。デフォルト値は `VLAN ID 1` です。有効な値は 0 - 4094 です。デフォルトの VLAN ID は、PVID (Port VLAN Identifier) とも呼ばれます。

基礎となるリンクのデフォルトの VLAN 値に一致する VLAN ID を持つタグ付き VLAN および VLAN タグ付き VNIC リンクを作成することはできません。リンク上のすべてのタグなしパケットは、すでにデフォルトの VLAN (PVID) に関連付けられています。デフォルトの VLAN 値に等しい VLAN ID を持つタグ付き VLAN または VLAN タグ付き VNIC リンクを正常に作成するには、まず基本となるリンクの `default_tag` プロパティーを別の VLAN 値に変更する必要があります。

`default_tag=0` に設定すると、リンク上のすべてのタグなしパケットがどの VLAN にも関連付けられなくなります。結果として、1 - 4094 の任意の VLAN ID を持つ VLAN リンクを作成できます。エンドポイントで誤って PVID のタグが付けられた受信パケットはすべてドロップされる可能性があることに注意してください。この状況は、特定のリンク上の一部のエンドポイントが PVID に同意していない場合に発生します。リンク上のエンドポイントはすべて同じ PVID を使用する必要があり、トラフィックに PVID のタグを付けてはいけません。

`stp_priority`
STP と RSTP のポート優先度の値を設定します。この値は、ブリッジ上の推奨されるルートポートを決定するために使用されます。数値が小さいほど、優先度は高くなります。デフォルト値は `128` です。有効な値の範囲は 0 - 255 です。

`stp_cost`
リンクを使用するための STP と RSTP のコストを設定します。デフォルト値は `auto` です。この場合はリンク速度に基づいてコストが設定され、10Mbps に `100`、100Mbps に `19`、1Gbps に `4`、および 10Gbps に `2` が使用されます。有効な値の範囲は 1 - 65535 です。

`stp_edge`
ブリッジのエッジポート検出を有効または無効にします。`0` (false) に設定されていると、どのタイプのブリッジ PDU も認識されない場合でも、システムはポートがほかのブリッジに接続されているとみなします。デフォルト値は `1` で、エッジポートが自動的に検出されます。

`stp_p2p`
ブリッジのポイントツーポイント動作モードを設定します。指定可能な値は、`true`、`false`、および `auto` です。`auto` に設定すると、ポイントツーポイント接続が自動的に検出されます。`true` に設定すると、ポートモードで強制的にポイントツーポイントが使用されます。`false` に設定すると、ポートモードで強制的に通常のマルチポイントモードが使用されます。デフォルト値は `auto` です。

`stp_mcheck`
このリンク上で RSTP の `Force BPDU Migration Check` 手順を実行するようにシステムを起動します。この手順は、このプロパティー値を `1` に設定することによって開始されます。このプロパティーは、元の `0` に自動的にリセットされます。この値は、次の条件が満たされないかぎり設定できません。

- リンクがブリッジされている
- ブリッジがスパニングツリーによって保護されている
- ブリッジの `force-protocol` 値が少なくとも 2 (RSTP) である

デフォルトの値は 0 です。

`protection`
リンク保護の 1 つ以上のタイプを有効にします。有効な値は次のとおりです。

`mac-nospoof`
MAC アドレスのスプーフィング対策。送信パケットの発信元 MAC アドレスが、そのリンクの構成済み MAC アドレスに一致している必要があります。一致していないパケットはドロップされます。リンクがゾーンに属している場合、`mac-nospoof` を有効にすると、ゾーンの所有者によってそのリンクの MAC アドレスが変更されることを防止できます。

`ip-nospoof`
IP アドレスのスプーフィング対策。この保護タイプは、リンクプロパティー `allowed-ips` と連動します。

allowed-ips は、IP (IPv4 または IPv6) アドレスを含むリストです。このリストは、デフォルトでは空の状態です。このリストに暗黙的に含まれているアドレスには、RFC 2464 に準拠したリンクローカル IPv6 アドレス (リンクの MAC アドレスから派生)、DHCP 応答から学習された IPv4/IPv6 アドレス、および指定されない (すべて 0 の) IPv4/IPv6 アドレスがあります。

送信 IP パケットは、その発信元アドレスが allowed-ips に含まれている場合に通過できます。

送信 ARP パケットは、その送信側プロトコルアドレスが allowed-ips に含まれている場合に通過できます。

allowed-ips を 1 つ以上の IP アドレスのセットに設定することによってデータリンクが保護されている場、このセットに含まれていない IP アドレスを構成しようとしても失敗し、EPERM エラーがユーザーに返されます。さらに、このインタフェースは IP パケットの転送には使用できず、このインタフェース上で ipadm(1M) 転送プロパティーを設定しようとすると EPERM エラーが表示されます。

dhcp-nospoof

DHCP クライアント ID (DHCPv6 の DUID) およびハードウェアアドレスのスプーフィング対策。この保護タイプは、リンクプロパティー allowed-dhcp-cids と連動します。

allowed-dhcp-cids リスト内の項目は、/etc/default/dhcpagent ファイル内の CLIENT_ID フィールドと同じ方法で書式設定するようにしてください。唯一の違いは、DUID を指定するときに , (コンマ) の代わりに . (ピリオド) を使用することです。詳細については、dhcpagent(1M) を参照してください。

送信 DHCP (v4/v6) パケットは、次の条件が満たされた場合にのみ通過できます。

- allowed-dhcp-cids が構成されておらず、パケットタイプが次の場合。
  - DHCPv4 の場合、クライアント ID フィールドが構成済み MAC アドレスに一致している必要があります。
  - DHCPv6 の場合、DUID のタイプが 1 または 3 で、DUID のリンクレイヤーアドレス部が構成済み MAC アドレスに一致している必要があります。
- allowed-dhcp-cids が構成されていて、パケットタイプが次の場合。
  - DHCPv4 の場合、クライアント ID フィールドが、このリスト上のいずれかの ID または構成済み MAC アドレスに一致している必要があります。
  - DHCPv6 の場合、DUID フィールドがこのリスト上のいずれかの ID に一致している必要があるか、または DUID のタイプが 1 または 3 で、かつ DUID のリンクレイヤーアドレス部が構成済み MAC アドレスに一致している必要があります。

`restricted`
この保護によって、送信パケットのタイプが IPv4、IPv6、および ARP だけに制限されます。

`zone`
リンクが属しているゾーンを指定します。`dladm` ではこのプロパティーを一時的にしか変更できないため、`-t` オプションを指定する必要があります。ゾーンの割り当てがリブート後も持続するように変更するには、`zonecfg(1M)` を使用してください。指定可能な値は、システム上で現在実行されている任意の排他的 IP ゾーンで構成されます。デフォルトでは、ゾーンのバインドは `zonecfg(1M)` に従います。

WiFi リンクのプロパティー

次の WiFi リンクのプロパティーがサポートされています。あるプロパティーを特定の値に設定できるかどうかは、ドライバやハードウェアによって異なることに注意してください。

`channel`
使用するチャネルを指定します。このプロパティーは、IBSS モードにあるときに、特定の WiFi リンクによってのみ変更できます。デフォルト値や、許可される値の範囲は、規定のドメインごとに異なります。

`powermode`
WiFi リンクの電源管理モードを指定します。指定可能な値は、`off` (電源管理が無効)、`max` (最大限の省電力)、および `fast` (パフォーマンス依存の電源管理) です。デフォルトは `off` です。

`radio`
WiFi リンクの無線モードを指定します。指定可能な値は、`on` または `off` です。デフォルトは `on` です。

`speed`
WiFi リンクの固定速度を指定します (Mbps 単位)。設定可能な値のセットは、ドライバやハードウェアによって異なります (ただし、`show-linkprop` で表示されます)。一般的な速度は 1、2、11、および 54 です。デフォルトでは、固定速度はありません。

Ethernet リンクのプロパティー

`ieee802.3(5)` に記載されている次の MII プロパティーが、読み取り専用モードでサポートされています。

- `duplex`
- `state`
- `adv_autoneg_cap`
- `adv_10gfdx_cap`
- `adv_1000fdx_cap`
- `adv_1000hdx_cap`
- `adv_100fdx_cap`
- `adv_100hdx_cap`
- `adv_10fdx_cap`

- adv_10hdx_cap

各 adv_ プロパティー (たとえば adv_10fdx_cap) には、自動ネゴシエーションで使用されるパラメータを制御する、読み取り/書き込み用の対応する en_ プロパティー (たとえば en_10fdx_cap) もあります。電源管理が行われていない場合、速度/二重通信の adv* パラメータによって、ネゴシエーション済みで、かつハードウェアで現在有効な値が提供されます。ただし、電源管理が有効になっている場合には、ハードウェアで現在公開されている速度/二重通信機能が、最初のリンクパラメータネゴシエーションで使用されたビットセットのサブセットになっている可能性があります。そのため、MII の adv_* パラメータは読み取り専用としてマークされており、最初のネゴシエーションで速度と二重通信のプロパティーを構成するための en_* パラメータセットが追加されています。

adv_autoneg_cap には、対応するプロパティー en_autoneg_cap がないことに注意してください。adv_autoneg_cap は自動ネゴシエーション自体を有効または無効にする 0/1 スイッチであるため、電源管理の影響は受けません。

さらに、次の Ethernet プロパティーが報告されます。

flowctrl
: デバイスによって通知されるフロー制御モードを確立します。有効な入力は次のいずれかです。

  auto
  : デバイス上のフロー制御モードは動的に決められます。デバイスで設定された実際のフロー制御モードを表示するには、flowctrl-effective リンクプロパティーを確認します。

  no
  : 有効なフロー制御がありません。

  rx
  : 一時停止フレームを受信し、その受信した一時停止フレームに基づいて動作します。

  tx
  : 輻輳が発生すると相手に一時停止フレームを送信しますが、受信した一時停止フレームは無視します。

  pfc
  : 一時停止されるトラフィックの優先順位の値を含む一時停止フレームを送信します。一時停止フレームを受信し、優先順位の値がフレーム内に指定されている値であるトラフィックを操作します。

  bi
  : 双方向のフロー制御。

  この値の実際の設定は、デバイスやリンクパートナーによって許可されている機能により制約を受けることに注意してください。

`gvrp-timeout`
VID 通知ブロードキャスト間の待機時間を指定します (ミリ秒単位)。

`flowctrl-effective`
デバイスで設定された実際のフロー制御モード。`flowctrl` プロパティーが `auto` に設定されている場合、これはフロー制御モードが有効であることを表します。これは読み取り専用プロパティーです。

`mtu`
デバイスでサポートされているクライアントの最大 SDU (送信データ単位)。有効な範囲は 68 - 65536 です。

`ntcs`
デバイスでサポートされているトラフィッククラスの数です。DCB (Data Center Bridging) の拡張機能をサポートしているデバイスは、複数のトラフィッククラスをサポートできます。このプロパティーを使用すると、デバイスが DCB の拡張機能をサポートしているかどうかを判断できます。これは読み取り専用プロパティーです。

`pfcmap`
このプロパティーは、PFC (優先順位ベースのフロー制御) が有効になっている 802.1p の優先順位の値を表すために使用します。これは 8 ビットマスクであり、対応する優先順位に対して PFC が有効かどうかを個々のビットで示します。PFC が有効になっている優先順位では、輻輳が発生した場合にデバイスはその優先順位に対して一時停止フレームを送信します。これが関係しているのは、`ntcs` がゼロよりも大きく、`flowctrl-effective` が `pfc` である場合のみです。

`speed`
(読み取り専用) デバイスの動作速度 (Mbps 単位)。

`tagmode`
このリンクプロパティーは、リンク上で送信されているパケットに 802.1Q VLAN タグが挿入される条件を制御します。このプロパティーに割り当てることができるモード値は次の 2 つです。

`normal` 次の条件のときに、送信パケットに VLAN タグを挿入します。
- そのパケットが VLAN に属している。
- ユーザーが優先度のタグ付けを要求しました。

`vlanonly` 送信パケットが VLAN に属している場合にのみ VLAN タグを挿入します。このモードでタグが挿入されているときにユーザーが 0 以外の優先度も要求した場合は、その優先度が尊重されて VLAN タグに含まれます。

デフォルト値は `vlanonly` です。

`vlan-announce`
このプロパティーは、VLAN ID の自動通知を制御します。有効になっている場合、デバイス上で構成されているすべての VNIC または VLAN の VID をブロード

キャストします。このプロパティーは、物理リンクと集積体の両方をサポートします。指定可能な値は、次のとおりです。

`off`
VID 通知は送信されません。

`gvrp`
802.1D で定義されている GVRP プロトコルを使用して通知が送信されます。ブロードキャストの頻度の構成については、`gvrp-timeout` を参照してください。

InfiniBand リンクのプロパティー

次のプロパティーは、IB パーティションオブジェクトリンク上でのみサポートされています。

`linkmode`
IB パーティションデータリンク上のリンクトランスポートのサービスタイプを設定します。デフォルト値は `cm` です。有効な値は次のとおりです。

`cm`
接続モード。このモードでは 65520 のデフォルトの MTU を使用し、65535 バイトの最大 MTU をサポートします。リモートノードに対して接続モードが使用できない場合は、代わりに低信頼データグラムモードが自動的に使用されます。

`ud`
低信頼データグラムモード。このモードでは 2044 のデフォルトの MTU を使用し、4092 バイトの最大 MTU をサポートします。

IP トンネルリンクのプロパティー

次の IP トンネルリンクのプロパティーがサポートされています。

`hoplimit`
トンネルリンクのカプセル化している外側の IP ヘッダーの IPv4 TTL または IPv6 ホップ制限を指定します。このプロパティーは、すべてのトンネルタイプに対して存在します。デフォルトの値は 64 です。

`encaplimit`
RFC 2473 で定義されている IPv6 トンネルの IPv6 カプセル化制限を指定します。この値は、特定のトンネルパケットに対するトンネルのネスト制限です。デフォルトの値は 4 です。0 の値を指定すると、カプセル化制限が無効になります。

使用例

例 1　集積体の構成

キー 1 を使用してデバイス `bge0` (リンク名 `net0`) および `bge1` (リンク名 `net1`) の集積体上にデータリンクを構成するには、次のコマンドを入力します。

```
# dladm create-aggr -d net0 -d net1 1
```

例 2　WiFi リンクへの接続

システム上で使用できるセキュリティー保護されていない最適なネットワークに 1 つの WiFi リンクで (connect-wifi に対して指定された優先順位付け規則に従って) 接続するには、次のコマンドを入力します。

```
# dladm connect-wifi
```

例 3　WiFi キーの作成

WEP キー mykey を対話形式で作成するには、次のコマンドを入力します。

```
# dladm create-secobj -c wep mykey
```

あるいは、ファイルの内容を使用して WEP キー mykey を非対話形式で作成するには、次のコマンドを入力します。

```
# umask 077
 # cat >/tmp/mykey.$$ <<EOF
 12345
 EOF
 # dladm create-secobj -c wep -f /tmp/mykey.$$ mykey
 # rm /tmp/mykey.$$
```

例 4　暗号化された指定の WiFi リンクへの接続

キー mykey を使用してリンク ath0 上の ESSID wlan に接続するには、次のコマンドを入力します。

```
# dladm connect-wifi -k mykey -e wlan ath0
```

例 5　リンクのプロパティーの変更

リンク pcwl0 上で powermode を値 fast に設定するには、次のコマンドを入力します。

```
# dladm set-linkprop -p powermode=fast pcwl0
```

例 6　WPA で保護された WiFi リンクへの接続

WPA キー psk を作成し、次のコマンドを入力します。

```
# dladm create-secobj -c wpa psk
```

次に、キー psk を使用してリンク ath0 上の ESSID wlan に接続するには、次のコマンドを入力します。

```
# dladm connect-wifi -k psk -e wlan ath0
```

例 7　リンクの名前の変更

bge0 リンクの名前を mgmt0 に変更するには、次のコマンドを入力します。

```
# dladm rename-link bge0 mgmt0
```

例 8 ネットワークカードの交換

前の例で示した mgmt0 というリンク名の bge0 デバイスを、ハードウェア障害のために ce0 デバイスに交換する必要があるとします。bge0 NIC を物理的に取り外し、新しい ce0 NIC に交換します。新しく追加した ce0 デバイスを、以前に bge0 に関連付けられていた mgmt0 構成に関連付けるには、次のコマンドを入力します。

```
# dladm rename-link ce0 mgmt0
```

例 9 ネットワークカードの取り外し

前の例で、bge0 NIC を別の NIC に交換するのではなく、ハードウェアを取り外すだけで交換は行わない場合を考えてみます。この場合、mgmt0 のデータリンク構成を、前の例で示したように別の物理デバイスに関連付けるのではなく、削除する必要があります。物理ハードウェア (この場合は bge0) が取り外された mgmt0 データリンクに関連付けられたデータリンク構成を削除するには、次のコマンドを入力します。

```
# dladm delete-phys mgmt0
```

例 10 解析可能な出力による単一のフィールドの取得

次の割り当てによって、リンク net0 の MTU が mtu という名前の変数に保存されます。

```
# mtu=‘dladm show-link -p -o mtu net0‘
```

例 11 解析可能な出力によるリンクの状態表示

次のスクリプトは、システム上の各リンクの状態を表示します。

```
# dladm show-link -p -o link,state | while IFS=: read link state; do
            print "Link $link is in state $state"
        done
```

例 12 VNIC の構成

1 つの物理リンク bge0 上に hello0 および test1 という名前の 2 つの VNIC を作成します。

```
# dladm create-vnic -l bge0 hello0
# dladm create-vnic -l bge0 test1
```

例 13 VNIC の構成、および帯域幅と優先度の割り当て

1 つの物理リンク bge0 上に hello0 および test1 という名前の 2 つの VNIC を作成し、hello0 を、ファクトリ MAC アドレスと 50Mbps の最大帯域幅を持つ高優先度の VNIC にします。test1 を、ランダムな MAC アドレスと 100Mbps の最大帯域幅を持つ低優先度の VNIC にします。

```
# dladm create-vnic -l bge0 -m factory -p maxbw=50,priority=high hello0
# dladm create-vnic -l bge0 -m random -p maxbw=100M,priority=low test1
```

例 14 ファクトリ MAC アドレスを持つ VNIC の構成

最初に、使用可能なファクトリ MAC アドレスを一覧表示し、いずれかのアドレスを選択します。

```
# dladm show-phys -m bge0
LINK            SLOT         ADDRESS              INUSE    CLIENT
net0            primary      0:e0:81:27:d4:47     yes      net0
net0            1            8:0:20:fe:4e:a5      no
net0            2            8:0:20:fe:4e:a6      no
net0            3            8:0:20:fe:4e:a7      no
```

hello0 という名前の VNIC を作成し、スロット 1 のアドレスを使用します。

```
# dladm create-vnic -l bge0 -m factory -n 1 hello0
# dladm show-phys -m bge0
LINK            SLOT         ADDRESS              INUSE    CLIENT
net0            primary      0:e0:81:27:d4:47     yes      net0
net0            1            8:0:20:fe:4e:a5      yes      hello0
net0            2            8:0:20:fe:4e:a6      no
net0            3            8:0:20:fe:4e:a7      no
```

例 15 ユーザー指定の MAC アドレスを持つ VNIC の作成、およびプロセッサのセットへのバインド

ユーザー指定の MAC アドレスと 0、2、4-6 のプロセッサバインドを持つ、VNIC という名前の hello0 を作成します。

```
# dladm create-vnic -l bge0 -m 8:0:20:fe:4e:b8 -p cpus=0,2,4-6 hello0
```

例 16 物理 NIC のない仮想ネットワークの作成

最初に、stub1 という名前の etherstub を作成します。

```
# dladm create-etherstub stub1
```

etherstub 上に hello0 と test1 という名前の 2 つの VNIC を作成します。この操作によって、hello0 と test1 を接続する仮想スイッチが暗黙的に作成されます。

```
# dladm create-vnic -l stub1 hello0
# dladm create-vnic -l stub1 test1
```

例 17 ブリッジ情報の表示

次の各コマンドは、オプションを指定しない場合とさまざまなオプションを指定した場合の show-bridge サブコマンドを使用しています。

```
# dladm show-bridge
BRIDGE       PROTECT ADDRESS           PRIORITY DESROOT
foo          stp     32768/8:0:20:bf:f 32768    8192/0:d0:0:76:14:38
bar          stp     32768/8:0:20:e5:8 32768    8192/0:d0:0:76:14:38

# dladm show-bridge -l foo
```

例 17 ブリッジ情報の表示 (続き)

```
LINK         STATE        UPTIME   DESROOT
hme0         forwarding   117      8192/0:d0:0:76:14:38
qfe1         forwarding   117      8192/0:d0:0:76:14:38

# dladm show-bridge -s foo
BRIDGE       DROPS        FORWARDS
foo          0            302

# dladm show-bridge -ls foo
LINK         DROPS     RECV      XMIT
hme0         0         360832    31797
qfe1         0         322311    356852

# dladm show-bridge -f foo
DEST              AGE     FLAGS  OUTPUT
8:0:20:bc:a7:dc   10.860  --     hme0
8:0:20:bf:f9:69   --      L      hme0
8:0:20:c0:20:26   17.420  --     hme0
8:0:20:e5:86:11   --      L      qfe1
```

例 18 IPv4 トンネルの作成

次の一連のコマンドは、66.1.2.3 と 192.4.5.6 の間に `mytunnel0` という名前の永続的な IPv4 トンネルリンクを作成してから、そのリンクを表示しています。

```
# dladm create-iptun -T ipv4 -a local=66.1.2.3,remote=192.4.5.6 mytunnel0
# dladm show-iptun mytunnel0
LINK            TYPE  FLAGS  SOURCE              DESTINATION
mytunnel0       ipv4  --     66.1.2.3            192.4.5.6
```

このトンネルリンク上に、ポイントツーポイント IP インタフェースを作成できます。

```
# ipadm create-ip mytunnel0
# ipadm create-addr -T static -a local=10.1.0.1,remote=10.1.0.2 \
mytunnel0/addr
# ipadm show-addr mytunnel0/addr
ADDROBJ           TYPE     STATE        ADDR
mytunnel0/addr    static   ok           10.1.0.1->10.1.0.2
```

例 19 6to4 トンネルの作成

次のコマンドは、6to4 トンネルリンクを作成します。6to4 ルーターの IPv4 アドレスは 75.10.11.12 です。

```
# dladm create-iptun -T 6to4 -a local=75.10.11.12 sitetunnel0
# dladm show-iptun sitetunnel0
LINK            TYPE  FLAGS  SOURCE              DESTINATION
```

例 19　6to4 トンネルの作成　　(続き)

```
sitetunnel0     6to4  --     75.10.11.12         --
```

次のコマンドは、このトンネル上に IPv6 インタフェースを作成します。

```
# ipadm create-ip sitetunnel0
# ipadm show-addr sitetunnel0/_a
ADDROBJ           TYPE     STATE        ADDR
sitetunnel0/_a    static   ok           2002:4b0a:b0c::1/16
```

システムは 6to4 IP インタフェース上に IPv6 アドレスを自動的に構成することに注意してください。6to4 トンネルリンク上で IPv6 アドレスが構成される方法については、ipadm(1M) を参照してください。

例 20　リンク保護の使用

リンク保護を有効にするには:

```
# dladm set-linkprop \
-p protection=mac-nospoof,restricted,ip-nospoof,dhcp-nospoof vnic0
```

リンク保護を無効にするには:

```
# dladm reset-linkprop -p protection vnic0
```

allowed-ips リストを変更するには:

```
# dladm set-linkprop -p allowed-ips=10.0.0.1,10.0.0.2 vnic0
```

allowed-dhcp-cids リストを変更するには:

```
# dladm set-linkprop -p allowed-dhcp-cids=hello vnic0
```

結果の構成を表示するには:

```
# dladm show-linkprop -p protection,allowed-ips vnic0

LINK     PROPERTY         PERM   VALUE        DEFAULT   POSSIBLE
vnic0    protection       rw     mac-nospoof, --        mac-nospoof,
                                restricted,            restricted,
                                ip-nospoof,            ip-nospoof,
                                dhcp-nospoof           dhcp-nospoof

vnic0    allowed-ips      rw     10.0.0.1,    --        --
                                10.0.0.2

vnic0    allowed-dhcp-cids rw    hello        --        --
```

例 21 IB パーティションの作成

次のコマンドは、物理リンク ibp0 上にパーティションキー 0xffff を持つパーティション ffff.ibp0 を作成します。

```
# dladm create-part -P ffff -l ibp0 ffff.ibp0
```

例 22 IB パーティション情報の表示

次のコマンドは、IB パーティション情報を表示します。

```
# dladm show-part
LINK         PKEY OVER         STATE    FLAGS
ffff.ibp0    FFFF ibp0         up       ----
```

例 23 IB データリンク情報の表示

次のコマンドは、IB データリンク情報を表示します。

```
# dladm show-ib
LINK         HCAGUID         PORTGUID        PORT STATE  PKEYS
ibp0         3BA000100CD7C   3BA000100CD7D   1    down   FFFF
ibp1         3BA000100CD7C   3BA000100CD7E   2    down   FFFF
ibp3         5AD0000033634   5AD0000033636   2    up     FFFF,8001
ibp2         5AD0000033634   5AD0000033635   1    up     FFFF,8001
```

例 24 パーティションの削除

次のコマンドは、パーティション ffff.ibp0 を削除します。

```
# dladm delete-part ffff.ibp0
```

例 25 show-link によるパーティション情報の表示

次のコマンドは、show-link サブコマンドを使用してパーティション情報を表示します。

```
# dladm show-link
LINK        CLASS     MTU    STATE    OVER
e1000g0     phys      1500   up       --
e1000g1     phys      1500   unknown  --
ibp0        phys      65520  down     --
ibp3        phys      65520  up       --
ibp2        phys      65520  up       --
ibp1        phys      65520  down     --
pffff.ibp0  part      2044   down     ibp0
p8001.ibp2  part      65520  unknown  ibp2
```

例 26 大域ゾーンからのすべてのゾーン内のリンクの表示

次に示す show-link コマンドは、大域ゾーンからすべてのゾーン内のデータリンクを表示します。大域ゾーンに含まれないリンクは、ゾーン名の接頭辞のあとにスラッシュ (/) 区切り文字を付けて表示されます。

例 26 大域ゾーンからのすべてのゾーン内のリンクの表示 (続き)

この例では、net0 は大域ゾーン内で作成された VNIC であり、zone1/net0 は zone1 に対して自動的に作成された VNIC であり、zone2/net0 は zone2 に対して自動的に作成された VNIC です。

```
# dladm show-link
LINK                CLASS     MTU    STATE    OVER
e1000g0             phys      1500   up       --
e1000g1             phys      8170   unknown  --
e1000g2             phys      1500   unknown  --
e1000g3             phys      1500   unknown  --
net0                vnic      1500   up       e1000g0
zone1/net0          vnic      1500   up       e1000g0
zone2/net0          vnic      1500   up       e1000g0
```

例 27 大域ゾーン内のリンクの表示

次の show-link コマンドは、大域ゾーン内のデータリンクのみを表示します。

```
# dladm show-link -z global
LINK                CLASS     MTU    STATE    OVER
e1000g0             phys      1500   up       --
e1000g1             phys      8170   unknown  --
e1000g2             phys      1500   unknown  --
e1000g3             phys      1500   unknown  --
net0                vnic      1500   up       e1000g0
```

例 28 指定されたゾーンのリンクの表示

次の show-link コマンドは、特定の非大域ゾーン内のデータリンクを表示します。

```
# dladm show-link -z zone1
LINK                CLASS     MTU    STATE    OVER
zone1/net0          vnic      1500   up       e1000g0
```

例 29 大域ゾーンからの指定されたゾーンのリンクの表示

次の show-link コマンドは、特定の非大域ゾーン内のデータリンクを大域ゾーンから表示します。

```
# dladm show-link -z zone1
LINK                CLASS     MTU    STATE    OVER
zone1/net0          vnic      1500   up       e1000g0
```

例 30 非大域ゾーン内のリンクの表示

次に示す show-link は、zone1 から呼び出され、そのゾーンのデータリンクのみを表示します。

例 30 非大域ゾーン内のリンクの表示 (続き)

show-link の出力では、zone1/ 接頭辞は表示されません。接頭辞が表示されないのは、このコマンドがゾーンの内部から呼び出されたからです。

```
# zlogin zone1
# dladm show-link -z zone1
LINK                CLASS     MTU    STATE    OVER
net0                vnic      1500   up       ?
```

例 31 -Z オプションを使用した現在のゾーンの表示

下記のコマンドでは、次の条件を前提としています。

- リンク net1 は現在 zoneA に割り当てられています。エントリ net1 および zoneA/net1 は同じリンクを表しています。これらの 2 つのエントリの ZONE 列は同じで、リンクが現在割り当てられているゾーンの名前になります。
- リンク net2 はどの非大域ゾーンにも割り当てられていません。
- リンク zoneB/net2 は、zoneB 用に作成された自動 VNIC です。
- リンク zoneC/net2 は、zoneC 用に作成された自動 VNIC です。
- リンク zoneD/net2 は、zoneD の内側で作成された IP トンネルです。net1 の場合と異なり、net2 の各エントリは異なるリンクを表します。これらのエントリの ZONE 列は異なります。

```
# dladm show-link -Z
LINK        ZONE      CLASS     MTU    STATE    OVER
e1000g0     global    phys      1500   up       --
e1000g1     global    phys      1500   up       --
net1        zoneA     vnic      1500   up       e1000g0
zoneA/net1  zoneA     vnic      1500   up       e1000g0
net2        global    vnic      1500   up       e1000g1
zoneB/net2  zoneB     vnic      1500   up       e1000g1
zoneC/net2  zoneC     vnic      1500   up       e1000g1
zoneD/net2  zoneD     iptun     65515  up       --
```

属性

次の属性については、attributes(5) を参照してください。

/usr/sbin

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

/sbin

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

/usr/sbin と /sbin の両方について、show-aggr、show-link、および show-vnic サブコマンドの -s オプションと -i オプションは「確実/廃止」です。

/usr/sbin と /sbin の両方について、show-linkprop の *effective プロパティーはインタフェースの安定性が「流動的」です。

前述の dladm サブコマンドで説明されているブリッジ関連のサブコマンドを使用するには pkg://solaris/network/bridging パッケージのインストールが必要です。

関連項目

acctadm(1M), autopush(1M), dhcpagent(1M), dlstat(1M), ifconfig(1M), ipadm(1M), ipsecconf(1M), ndd(1M), pooladm(1M), poolcfg(1M), psrset(1M), vrrpadm(1M), wpad(1M), zonecfg(1M), attributes(5), ieee802.3(5), dlpi(7P)

注意事項

集積体のサブコマンドで集積体を参照する場合は、リンク名で参照することをお勧めします。整数キーによる集積体の参照は、下位互換性のためにサポートされていますが、必要ではありません。集積体を作成するときにリンク名の代わりにキーが指定されている場合は、その集積体のリンク名が dladm によって aggr*key* として自動的に生成されます。

名前

dlmgmtd – データリンク管理デーモン

形式

```
/usr/sbin/dlmgmtd
```

```
svc:/network/datalink-management:default
```

機能説明

dlmgmtd は、ネットワークデータリンクインタフェースの管理イベントを処理するシステムデーモンです。このデーモンは、サービス管理機能 (SMF) サービスインスタンスによって制御されます。

```
svc:/network/datalink-management:default
```

このデーモンは、直接起動しないでください。このデーモンは、管理インタフェースやプログラミングインタフェースの構成要素ではありません。データリンクを管理するための管理インタフェースは dladm(1M) です。

オプション

このデーモンにオプションはありません。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	非公開

関連項目

dladm(1M), attributes(5)

名前

dlstat – データリンクの統計情報の報告

形式

```
dlstat [-r | -t] [-Z] [-i interval] [-z zone[,...]] [link]

dlstat [-a | -A] [-Z] [-i interval] [-p] [ -o field[,...]]
     [-u R|K|M|G|T|P] [-z zone[,...]] [link]

dlstat show-phys [-r | -t] [-Z] [-i interval] [-a]
     [-p] [-o field[,...]] [-u R|K|M|G|T|P] [-z zone[,...]] [link]

dlstat show-link [-r | -t] [-Z] [-i interval] [-a]
     [-p] [-o field[,...]] [-u R|K|M|G|T|P] [-z zone[,...]] [link]

dlstat show-link -h [-Z] [-a] -f filename [-d] [-F format] [-s time]
          [-e time] [-z zone[,...]] [link]

dlstat show-aggr [-r | -t] [-Z] [-i interval] [-p] [-o field[,...]]
     [-u R|K|M|G|T|P] [-z zone[,...]] [link]
```

機能説明

`dlstat` コマンドは、データリンクに関する実行時統計情報を報告します。`dladm(1M)` `show-phys` は、`dlstat show-phys` にリンク名情報を提供します。`dladm(1M)` `show-link` は、`dlstat show-link` にリンク名情報を提供します。`dladm(1M)` `show-aggr` は、`dlstat show-aggr` にリンク集積体情報を提供します。

`dlstat` には、前述の「形式」に示されている 6 つの形式のコマンドがあります。最初の 2 つの形式にはサブコマンドがないのに対し、残りの形式にはあります。6 つの形式はすべて、後述の「サブコマンド」で説明されています。

オプション

`dlstat` コマンドには次のオプションとオペランドがあります。これらは、(特に明示的にマークされていないかぎり) 後述の「サブコマンド」に示されているいくつかのコマンド形式の間で共通です。

`-a`
統計情報フィールド全体をすべてダンプします。

`-i` *interval*
統計情報が更新される間隔を秒単位で指定します。デフォルトの間隔は 1 秒です。

`-o` *field*[,...]
表示する出力フィールドのコンマ区切りリストを表示します (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはサポートされるすべてのフィールドを表示するための特殊な値 `all` にする必要があります。

サポートされる RX フィールドは次のとおりです。

- `link`
- `index`
- `rbytes`
- `ipkts`
- `intrs` (サブコマンドなしで、または `show-link` とともにのみ適用可能)

- polls (サブコマンドなしで、または show-link とともにのみ適用可能)
- idrops (サブコマンドなしで、または show-link とともにのみ適用可能)

TX 側のフィールドは次のとおりです。

- link
- index
- obytes
- opkts
- odrops (サブコマンドなしで、または show-link とともにのみ適用可能)

show-phys サブコマンドでは、index カラムは物理デバイス内の個々の RX および TX ハードウェアリングを識別します。show-link サブコマンドでは、index カラムはデータリンク内の RX および TX ハードウェアレーンを識別します。ハードウェアリングとハードウェアレーンの違いについての説明は、『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』を参照してください。

-p
安定したマシン解析可能な形式を使用して出力します。

-r
受信側の統計情報のみを表示します。受信されたバイト数とパケット数や、ハードウェアとソフトウェアのドロップなどが含まれます。完全な一覧については、「使用例」を参照してください。

-t
送信側の統計情報のみを表示します。送信されたバイト数とパケット数や、ドロップなどが含まれます。完全な一覧については、「使用例」を参照してください。

-u R|K|M|G|T|P
使用した場合、すべての統計情報の表示単位を選択できます (たとえば、R: 生カウント、K: キロビット、M: メガビット、T: テラビット、P: ペタビット)。使用しない場合、必要に応じて異なる単位が使用され、*xy.zU* 形式で統計情報が表示されます (*x*、*y*、および *z* は数値、*U* は該当する単位です)。

-Z
ZONE 列を出力に表示します。

-z *zone*[,...]
指定されたゾーン内のリンクのみの統計情報を表示します。デフォルトでは、大域ゾーンから実行されている場合、dlstat はすべてのゾーン内のリンクの統計情報を表示します。

非大域ゾーンから実行されている場合、dlstat はそのゾーン内のリンクのみの統計情報を表示します。非大域ゾーンでは、ほかのゾーン内のリンクを表示することはできません。

link
使用した場合、指定のリンク、物理デバイス (`show-phys` の場合)、または集積体 (`show-aggr` の場合) のみの統計情報を表示します。それ以外の場合は、すべてのリンク、デバイス、または集積体の統計情報を表示します。

サブコマンド

`dlstat` は、次のコマンド形式をサポートします。

`dlstat` [`-r`|`-t`] [`-Z`] [`-i` *interval*] [`-z` *zone*[,...]] [*link*]
すべてのリンクを繰り返し検査して、統計情報を報告します。出力は、リンク使用率の高い順にソートされます。リンクが指定されていない場合、システムはすべてのリンクの統計情報を表示します。トラフィックの統計情報は、物理デバイスごとにではなく、リンクごとに表示されます。たとえば、ある物理リンク上に構成されている VNIC について、その VNIC を通過するトラフィックは、ベースとなる物理リンクの統計情報には反映されません。ただし、リンクの統計情報には、そのリンク上に構成されたユーザー定義のフローに一致するトラフィックが含まれます。

このコマンド形式には、前述の「オプション」で説明されていないオプションが 1 つあります。

`-A`
このデータリンクのすべての統計情報フィールドをダンプします。このコマンドの出力統計情報には、ほかのすべての `dlstat` コマンドによって報告されるすべての統計情報が含まれます。

`dlstat` [`-a`|`-A`] [`-Z`] [`-i` *interval*] [`-p`] [`-o` *field*[,...]] [`-u` R|K| M|G|T|P] [`-z` *zone*[,...]] [*link*]
どの統計情報を表示するかを指定できます。

このコマンド形式のオプションは、前述の「オプション」で説明されています。

`dlstat show-phys` [`-r`|`-t`] [`-Z`] [`-i` *interval*] [`-a`] [`-p`] [`-o` *field*[,...]] [`-u` R|K|M|G|T|P] [`-z` *zone*[,...]] [*link*]
物理デバイスの統計情報を表示します。

このサブコマンドのオプションは、前述の「オプション」で説明されています。

`dlstat show-link` [`-r`|`-t`] [`-Z`] [`-i` *interval*] [`-a`] [`-p`] [`-o` *field*[,...]] [`-u` R|K|M|G|T|P] [`-z` *zone*[,...]] [*link*]
リンクの統計情報を表示します。

`dlstat show-link -h` [`-Z`] [`-a`] `-f` *filename* [`-d`] [`-F` *format*] [`-s` *time*] [`-e` *time*] [`-z` *zone*[,...]] [*link*]
格納されている拡張アカウンティングファイルからネットワーク使用率の履歴を表示します。この構文を使用するには、`acctadm(1M)` によってネットワークアカウンティングがあらかじめ構成され、有効になっている必要があります。デフォルトの出力は、拡張アカウンティングが有効になっていた全期間における既存のリンクのネットワーク使用率の要約です。

link 引数は、前述の「オプション」で説明されているとおりです。

`-a`
拡張アカウンティングが有効になっている場合に、特定の期間におけるすべてのネットワーク使用率の履歴を表示します。これには、すでに削除されたリンクに関する使用率の情報が含まれます。

`-f` *filename*
ネットワーク使用率の履歴の拡張アカウンティングレコードが読み取られるファイルを指定します。

`-d`
ロギング情報が存在する日付を表示します。日付は、*mm*/*dd*/ *yyyy* の形式で表示されます。

`-F` *format*
ネットワーク使用率の履歴情報の出力形式を指定します。`gnuplot` が唯一のサポートされる形式です。

`-s` *time*
`-e` *time*
データ表示の開始時刻と停止時刻を指定します。時刻は、*MM*/*DD*/ YYYY,hh:mm *ss* の形式で指定します。*hh* は 24 時間表記を使用します。

`dlstat show-aggr` [`-r` | `-t`] [`-Z`] [`-i` *interval*] [`-p`] [`-o` *field*[,...]] [`-u` R |K|M|G|T|P] [`-z` *zone*[,...]] [*link*]
1 つの集積体についてポートごとの統計情報を表示します。

このサブコマンドのオプションは、前述の「オプション」で説明されています。

使用例

例 1　統計情報の表示

すべてのリンクの統計情報を表示するには、次のコマンドを入力します。統計情報は、3 桁の数値 + 小数点 + 2 桁の数値に、該当する単位が付けられた形で表示されます。

```
# dlstat
      LINK   IPKTS  RBYTES   OPKTS  OBYTES
   e1000g0 101.88K  32.86M  40.16K   4.37M
     nxge1   4.50M   6.78G   1.38M  90.90M
     vnic1       8     336       0       0
      net0  73.96K   6.81M       0       0
zone1/net0 144.47K  13.32M     247  16.29K
zone2/net0 132.89K  12.25M     236  15.82K
```

例 2　RX 側の統計情報の表示

次のコマンドは、1 秒ごとの受信側の統計情報を表示します。

```
# dlstat -r -i 1
    LINK   IPKTS  RBYTES   INTRS   POLLS   IDROPS
e1000g0 101.91K  32.86M  87.56K  14.35K         0
  nxge1   9.61M  14.47G   5.79M   3.82M         0
```

例 2　RX 側の統計情報の表示　　(続き)

```
  vnic1       8     336       0       0         0
e1000g0       0       0       0       0         0
  nxge1  82.13K 123.69M  50.00K  32.13K         0
  vnic1       0       0       0       0         0
      .       .       .       .       .         .
      .       .       .       .       .         .
```

例 3　ハードウェアリングごとの統計情報の表示

次のコマンドは、受信側のハードウェアリングごとの統計情報を表示します。

```
# dlstat show-phys -r nxge1
 LINK TYPE   INDEX   IPKTS  RBYTES
nxge1   rx       0       0       0
nxge1   rx       1       0       0
nxge1   rx       2   1.73M   2.61G
nxge1   rx       3       0       0
nxge1   rx       4   8.44M  12.71G
nxge1   rx       5   5.68M   8.56G
nxge1   rx       6   4.90M   7.38G
nxge1   rx       7       0       0
```

例 4　レーンごとの統計情報の表示

次のコマンドは、受信側のレーンごとの統計情報を表示します。最初に、専用のハードウェアレーンを備えたインタフェースが表示されます。

```
# dlstat show-link -r nxge1
 LINK TYPE     ID INDEX   IPKTS  RBYTES   INTRS   POLLS  IDROPS
nxge1   rx  local    --       0       0       0       0       0
nxge1   rx     hw     1       0       0       0       0       0
nxge1   rx     hw     2   1.73M   2.61G   1.33M 400.22K       0
nxge1   rx     hw     3       0       0       0       0       0
nxge1   rx     hw     4   8.44M  12.71G   4.35M   4.09M       0
nxge1   rx     hw     5   5.68M   8.56G   3.72M   1.97M       0
nxge1   rx     hw     6   4.90M   7.38G   3.11M   1.80M       0
nxge1   rx     hw     7       0       0       0       0       0
```

次に、専用のハードウェアレーンがなく、ソフトウェアレーンのみを備えたインタフェースが表示されます。

```
# dlstat show-link -r ixgbe0
  LINK TYPE     ID INDEX   IPKTS  RBYTES   INTRS   POLLS IDROPS
ixgbe0   rx  local    --       0       0       0       0      0
ixgbe0   rx     sw    -- 794.28K   1.19G 794.28K       0      0
```

例 5　送信側の統計情報の表示

次のコマンドは、送信側の統計情報を 5 秒間隔で表示します。

例 5　送信側の統計情報の表示　　(続き)

```
# dlstat -t -i 5
  LINK   OPKTS  OBYTES  ODROPS
e1000g0  40.24K   4.37M      0
  nxge1   9.76M 644.14M      0
  vnic1       0       0      0
e1000g0       0       0      0
  nxge1  26.82K   1.77M      0
  vnic1       0       0      0
      .       .       .      .
      .       .       .      .
      .       .       .      .
```

例 6　送信側のリングの統計情報の表示

次のコマンドは、送信側のハードウェアリングの統計情報を表示します。

```
# dlstat show-phys -t nxge1
 LINK TYPE INDEX   OPKTS  OBYTES
nxge1   tx     0      44   3.96K
nxge1   tx     1       0       0
nxge1   tx     2   1.48M 121.68M
nxge1   tx     3   2.45M 201.11M
nxge1   tx     4   1.47M 120.82M
nxge1   tx     5       0       0
nxge1   tx     6   1.97M 161.57M
nxge1   tx     7   4.59M 376.21M
nxge1   tx     8   2.43M 199.24M
nxge1   tx     9       0       0
nxge1   tx     10  3.23M 264.69M
nxge1   tx     11  1.88M 153.96M
```

例 7　送信側のレーンの統計情報の表示

次のコマンドは、送信側のレーンの統計情報を表示します。

```
# dlstat show-link -t nxge1
 LINK TYPE     ID INDEX   OPKTS  OBYTES  ODROPS
nxge1   tx     hw     0      32   1.44K       0
nxge1   tx     hw     1       0       0       0
nxge1   tx     hw     2   1.48M  97.95M       0
nxge1   tx     hw     3   2.45M 161.87M       0
nxge1   tx     hw     4   1.47M  97.25M       0
nxge1   tx     hw     5       3     276       0
nxge1   tx     hw     6   1.97M 130.05M       0
nxge1   tx     hw     7   4.59M 302.80M       0
nxge1   tx     hw     8   2.42M 302.80M       0
nxge1   tx     hw     9       0       0       0
nxge1   tx     hw     10  3.23M 213.05M       0
```

例 7 送信側のレーンの統計情報の表示 (続き)

```
nxge1   tx     hw     11  1.88M 123.93M       0
```

例 8 統計情報の特定のセットの選択

次のコマンドは、表示したい統計情報のセットを選択する方法を示しています。

```
# dlstat show-link -r -o LINK,TYPE,ID,INDEX,INTRS,POLLS nxge1
 LINK TYPE     ID INDEX   INTRS   POLLS
nxge1   rx  local    --       0       0
nxge1   rx  other    --       0       0
nxge1   rx     hw     1       0       0
nxge1   rx     hw     2   2.47M 753.90K
nxge1   rx     hw     3       0       0
nxge1   rx     hw     4   8.24M   7.72M
nxge1   rx     hw     5   6.96M   3.68M
nxge1   rx     hw     6   5.82M   3.36M
nxge1   rx     hw     7       0       0
```

例 9 ネットワーク使用率の履歴の表示

ネットワーク使用率の履歴の統計情報は、拡張アカウンティング機能 acctadm(1M) を使用すると格納できます。そのためには、次のようなコマンドを使用します。

```
# acctadm -e basic -f /var/log/net.log net
acctadm net
Network accounting: active
    Network accounting file: /var/log/net.log
  Tracked Network resources: basic
Untracked Network resources: src_ip,dst_ip,src_port,dst_port,protocol,
                             dsfield
```

その後、格納された履歴データは、次のようなコマンドを使用することで、要約形式で取得できます。

```
# dlstat show-link -h -f /var/log/net.log
LINK      DURATION  IPACKETS RBYTES      OPACKETS OBYTES      BANDWIDTH
e1000g0   80        1031     546908      0        0           2.44 Kbps
```

属性

次の属性についての詳細は、attributes(5) を参照してください。

/usr/sbin

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照

画面出力は「不確実」です。呼び出しは「確実」です。

関連項目

acctadm(1M), dladm(1M), ifconfig(1M), kstat(1M), netstat(1M), attributes(5)

名前

dmesg – エラーログを生成するためのシステム診断メッセージの収集

形式

```
/usr/bin/dmesg
```

```
/usr/sbin/dmesg
```

機能説明

dmesg は、システムエラーログの保守に関して、syslogd(1M) によって廃止されました。

dmesg は、最近出力された診断メッセージをシステムバッファーで見つけ出し、それを標準出力に書き込みます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

syslogd(1M), attributes(5)

名前

dminfo – report information about a device entry in a device maps file

形式

```
dminfo [-v] [-a] [-f pathname]
dminfo [-v] [-a] [-f pathname] -n dev -name...
dminfo [-v] [-a] [-f pathname] -d dev -path...
dminfo [-v] [-a] [-f pathname] -t dev -type...
dminfo [-v] [-a] [-f pathname] -u dm -entry
```

機能説明

`dminfo` reports and updates information about the `device_maps(4)` file.

オプション

The following options are supported

`-a`	Succeed if any of the requested entries are found. If used with `-v`, all entries that match the requested case(s) are printed.
`-d` *dev–path*	Search by *dev–path*. Search `device_maps(4)` for a device special pathname in the *device_list* field matching the *dev–path* argument. This option cannot be used with `-n`, `-t` or `-u`.
`-f` *pathname*	Use a device_maps file with *pathname* instead of `/etc/security/device_maps`.
`-n` *dev–name*	Search by *dev–name*. Search `device_maps(4)` for a *device_name* field matching *dev–name*. This option cannot be used with `-d`, `-t` or `-u`.
`-t` *dev–type*	Search by *dev–type*. Search `device_maps(4)` for a *device_type* field matching the given *dev–type*. This option cannot be used with `-d`, `-n` or `-u`.
`-u` *dm–entry*	Update the `device_maps(4)` file. This option is provided to add entries to the `device_maps(4)` file. The *dm–entry* must be a complete `device_maps(4)` file entry. The *dm–entry* has fields, as in the `device_maps` file. It uses the colon (`:`) as a field separator, and white space as the *device_list* subfield separators. The *dm–entry* is not made if any fields are missing, or if the *dm–entry* would be a duplicate. The default device maps file can be updated only by the super user.
`-v`	Verbose. Print the requested entry or entries, one line per entry, on the standard output. If no entries are specified, all are printed.

終了ステータス

0	Successful completion.
1	Request failed.
2	Incorrect syntax.

ファイル

`/etc/security/device_maps`

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

device_maps(4), attributes(5)

名前

dns-sd – マルチキャスト DNS (mDNS) および DNS サービス検出 (DNS-SD) テストツール

形式

`dns-sd -R` *name type domain port* `[`*key=value* `...]`

`dns-sd -B` *type domain*

`dns-sd -L` *name type domain*

`dns-sd -Q` *FQDN rrtype rrclass*

`dns-sd -C` *FQDN rrtype rrclass*

`dns-sd -P` *name type domain port host IP* `[`*key=value* `...]`

`dns-sd -E | -F | -A | -U | -N | -T | -M | -I`

機能説明

`dns-sd` コマンドは、`ping(1M)` や `traceroute(1M)` とほぼ同様のネットワーク診断ツールです。ただし、これらのツールとは異なり、その機能のほとんどが `dns-sd` 実行可能ファイル自体ではなく、すべてのアプリケーションで使用できるライブラリコード内に実装されています。`dns-sd` が使用しているライブラリ API は、`/usr/include/dns_sd.h` に記述されています。

`dns-sd` コマンドは対話形式での使用を主に想定しています。そのコマンド行引数や出力形式は変更される可能性があるため、シェルスクリプトから呼び出した場合の結果は予測できません。また、DNS サービス検出の非同期の性質のために、スクリプト指向のプログラミングにはあまり適していません。このような非同期の対話が最適に機能するアプリケーションは、マルチスレッドのアプリケーションか、またはメインのイベント処理ループを使用してキーストローク、ネットワークデータ、およびその他の非同期イベント通知をそれらが発生した時点で受信するアプリケーションです。

オプション

次のオプションがサポートされています。

`-R` *name type domain port* [*key=value* ...]

指定のドメイン内にあり、指定の *name* と *type* を持ち、(現在のマシン上にある) 指定の *port* で待機しているサービスを登録 (通知) します。

name には、任意の有効な Unicode 文字 (ドット、スペース、スラッシュ、コロンなど (制限はなし)) を含む、UTF-8 で最大 63 バイト長の任意の Unicode テキストを使用できます。

type は、「_app-proto._tcp」または「_app-proto._udp」の形式にする必要があります。ここで、「app-proto」は、`http://www.dns-sd.org` のサービスタイプ (RFC 2782) リンクに登録されているアプリケーションプロトコル名です。

domain は、このサービスを登録するドメインです。現在の実装では、ローカルマルチキャストドメイン「local」のみがサポートされています。将来は、有効な DNS 更新サーバー (RFC 2136) を含む任意のドメイン内での登録がサポートされる予定です。ドメイン「.」は「妥当なデフォルト値の選択」の同義語で、現時点では「local」を意味します。

port は 0 - 65535 の数値で、サービスが待機している TCP または UDP ポート番号です。ポート 0 上にサービスを登録すると、アプリケーションはサービスの非可用性を明示的に通知できます。

必要に応じて、サービスの追加の属性をキー/値のペアで記述できます。このペアは、通知されるサービスの DNS TXT レコード内に格納されます。利用可能ななキーと値は、`http://www.dns-sd.org` のサービスタイプ (RFC 2782) リンクにあるサービス登録に一覧表示されています。

`-B` *type domain*
domain 内のサービス *type* のインスタンスをブラウズします。

有効なタイプについては、`http://www.dns-sd.org` のサービスタイプ (RFC 2782) リンクを参照してください。ドメイン名を省略または「.」の使用は、「妥当なデフォルト値の選択」を意味します。

`-L` *name type domain*
指定されたサービスに接続して使用するために必要な情報を検索および表示します。この情報には、そのサービスが使用可能なマシンのホスト名、そのサービスが待機しているポート番号、およびそのサービスのプロパティーが記述されている TXT レコード属性 (もしあれば) が含まれます。

標準的なアプリケーションでは、ブラウズがまれにしか発生しないのに対し、検索 (または「解決」) はサービスが使用されるたびに発生します。たとえば、ユーザーがデフォルトのプリンタを選択するためにネットワークをブラウズすることはそれほど頻繁にはありませんが、デフォルトのプリンタがいったん選択されると、ユーザーが Cmd-P キーを押して出力するたびに、その指定されたサービスが現在の IP アドレスとポート番号に解決されます。

`-Q` *FQDN rrtype rrclass*
任意のリソースレコードタイプおよびクラスに対する汎用のクエリー。

`-C` *FQDN rrtype rrclass*
任意のリソースレコードタイプおよびクラスに対する汎用のクエリー。このオプションでは、クエリーからの各結果も再確認されます。レコードの再確認では、`mdnsd(1M)` に対してレコードの有効性を確認するよう指示します。レコードが有効でない場合、`mdnsd(1M)` はこのデーモンのキャッシュだけでなく、ネットワーク上のほかの `mdnsd(1M)` キャッシュからもそのレコードをフラッシュします。

`-P` *name type domain port host IP* [*key=value* ...]
指定のドメイン内にあり、指定の *name* と *type* を持ち、指定のポートで待機し、別のホスト上でアクセス可能なサービスを登録 (通知) します。このオプションは、別のホスト上でアクセス可能なサービスをプロキシによって通知する場合に使用するようにしてください。このサービスにアクセスするためのホスト名と IPv4 アドレスを指定する必要があります。

-E
推奨される登録ドメインを検出します。このオプションを使用すると、サービスを登録するための推奨ドメインが返されます。推奨される登録ドメインは、resolv.conf(4) でネームサーバーに対してクエリーを実行することによって返されます。

-F
推奨されるブラウズドメインを検出します。このオプションを使用すると、サービスをブラウズするための推奨ドメインが返されます。推奨されるブラウズドメインは、resolv.conf(4) でネームサーバーに対してクエリーを実行することによって返されます。

-A
マルチキャスト DNS へのサービスの登録をテストしたあと、マルチキャスト DNS への DNS レコードの追加、更新、および削除操作をテストします。

-U
マルチキャスト DNS へのサービスの登録をテストしたあと、マルチキャスト DNS に登録されたサービスの DNS TXT レコードの更新をテストします。

-N
マルチキャスト DNS に登録されたサービスの大きな NULL レコードの追加をテストします。

-T
マルチキャスト DNS に登録されたサービスの大きな TXT レコードの追加をテストします。

-M
複数の TXT レコードでの登録の作成をテストします。

-I
登録と、登録直後の TXT レコードの更新をテストします。

使用例

例 1 印刷サービスの通知

次のコマンドは、このマシンのポート 515 上に LPR 印刷サービスが存在することを通知し、DNS-SD と互換性がある印刷クライアントからこのサービスを使用できるようにします。

```
dns-sd -R "My Test" _printer._tcp. . 515 pdl=application/postscript
```

この登録を有効にするには、LPR サービスをポート 515 上で使用できるようにしてください。存在しないサービスを通知しても、あまり意味はありません。

例 2 Web ページの通知

次のコマンドは、Web ページがこのマシンのポート 80 上で HTTP サーバーから提供されていることを通知し、Safari やそのほかの DNS-SD と互換性がある Web クライアントの Bonjour リストにこのページが表示されるようにします。

例 2　Web ページの通知　　*(続き)*

```
dns-sd -R "My Test" _http._tcp . 80 path=/path-to-page.html
```

例 3　ローカルネットワーク上で通知される Web ページの検索

次のコマンドは、ローカルネットワーク上で通知される Web ページ (Safari で表示されるリストと同じもの) を検索します。

```
dns-sd -B _http._tcp
```

属性

次の属性についての詳細は、 attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/mdns
インタフェースの安定性	「流動的」

関連項目

mdnsd(1M), ping(1M), ping(1M), resolv.conf(4), attributes(5)

名前

dnssec-dsfromkey – DNSSEC DS RR 生成ツール

形式

```
dnssec-dsfromkey [-v level] [-1] [-2] [-a alg] keyfile

dnssec-dsfromkey -s [-v level] [-1] [-2] [-a alg] [-c class]
     [-d dir] keyfile
```

機能説明

dnssec-dsfromkey

オプション

次のオプションがサポートされています。

-1
ダイジェストアルゴリズムとして SHA-1 を使用します。デフォルトでは、SHA-1 と SHA-256 の両方を使用します。

-2
ダイジェストアルゴリズムとして SHA-256 を使用します。

-a *algorithm*
ダイジェストアルゴリズムを選択します。*algorithm* の値は、SHA-1 (SHA1) または SHA-256 (SHA256) のいずれかにする必要があります。これらの値は大文字と小文字が区別されません。

-v *level*
デバッグのレベルを設定します。

-s
鍵セットモード: 鍵ファイル名の代わりに、鍵セットファイルの DNS ドメイン名が引数になります。-c オプションと -d オプションは、このモードでのみ有効です。

-c *class*
DNS クラスを指定します (デフォルトは IN)。これは鍵セットモードでのみ有効です。

-d *directory*
directory ディレクトリ内の鍵セットファイルを検索します。鍵セットモードでないときは無視されます。

使用例

Kexample.com.+003+26160 の鍵ファイル名から SHA-256 DS RR を作成するには、次のようなコマンドを使用します。

```
# dnssec-dsfromkey -2 Kexample.com.+003+26160
```

このコマンドによって、次のような出力が生成されます。

```
example.com. IN DS 26160 5 2
3A1EADA7A74B8D0BA86726B0C227AA85AB8BBD2B2004F41A868A54F0
C5EA0B94
```

ファイル

鍵ファイルは、鍵 ID K*nnnn*.+*aaa*+*iiiii*、または完全なファイル名 K*nnnn*.+*aaa*+*iiiii*.key によって指定できます。これらの鍵は、dnssec-keygen(1M) によって生成されます。

鍵セットファイル名は、ディレクトリ、文字列 keyset-、および *dnsname* から作成されます。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/bind
インタフェースの安定性	「流動的」

関連項目

dnssec-keygen(1M), dnssec-signzone(1M), attributes(5)

RFC 3658、RFC 4509

『BIND 9 Administrator's Reference Manual』を参照してください。このマニュアルページの発行日付時点で、このドキュメントは https://www.isc.org/software/bind/documentation から利用できます。

注意

鍵ファイルのエラーにより、ファイルが存在する場合でも「ファイルが見つかりません」というメッセージが生成されることがあります。

名前

dnssec-keyfromlabel – DNSSEC 鍵生成ツール

形式

```
dnssec-keyfromlabel -a algorithm -l label [-c class] [-f flag] [-k]
      [-n nametype] [-p protocol] [-t type] [-v level] name
```

機能説明

`dnssec-keyfromlabel` は、指定されたラベルを含む鍵を暗号化ハードウェアデバイスから取得し、RFC 2535 および RFC 4034 で定義されている DNSSEC (Secure DNS) の鍵ファイルを作成します。

オプション

次のオプションがサポートされています。

`-a` *algorithm*
暗号化アルゴリズムを選択します。*algorithm* の値は、`RSAMD5` (RSA) または `RSASHA1`、`DSA`、`NSEC3RSASHA1`、`NSEC3DSA`、`DH` (Diffie-Hellman) のいずれかにする必要があります。これらの値は大文字と小文字が区別されません。

`DNSSEC` の場合、`RSASHA1` は実装が必須のアルゴリズムであるため、DSA が推奨されることに注意してください。また、`DH` では `-k` フラグが自動的に設定されることにも注意してください。

`-l` *label*
暗号化ハードウェア (PKCS#11) デバイス内の鍵のラベルを指定します。

`-n` *nametype*
鍵の所有者型を指定します。*nametype* の値は、`ZONE` (`DNSSEC` ゾーン鍵 (`KEY/DNSKEY`) の場合)、`HOST` または `ENTITY` (ホストに関連付けられた鍵 (`KEY`) の場合)、`USER` (ユーザーに関連付けられた鍵 (`KEY`) の場合)、`OTHER` (`DNSKEY`) のいずれかにする必要があります。これらの値は大文字と小文字が区別されません。

`-c` *class*
鍵を含む DNS レコードに、指定されたクラスが存在するべきであることを示します。指定されていない場合は、クラス `IN` が使用されます。

`-f` *flag*
指定されたフラグを `KEY/DNSKEY` レコードのフラグフィールドに設定します。認識されるフラグは `KSK` (Key Signing Key) `DNSKEY` のみです。

`-h`
`dnssec-keyfromlabel` のオプションと引数の簡単な要約を表示します。

`-k`
`DNSKEY` レコードではなく `KEY` レコードを生成します。

`-p` *protocol*
生成された鍵のプロトコル値を設定します。プロトコルは 0 - 255 の数値です。デフォルトは 3 (`DNSSEC`) です。この引数に指定可能なほかの値は、RFC 2535 およびそれ以降のバージョンに記載されています。

-t *type*
鍵の使用を示します。*type* は、AUTHCONF、NOAUTHCONF、NOAUTH、NOCONF のいずれかにする必要があります。デフォルトは AUTHCONF です。AUTH はデータを認証する機能を、CONF はデータを暗号化する機能を示します。

-v *level*
デバッグのレベルを設定します。

生成される鍵ファイル

dnssec-keyfromlabel が正常に完了すると、K*nnnn*.+*aaa*+*iiiii* の形式の文字列が標準出力に表示されます。これは生成された鍵ファイルの識別文字列で、次の内容を示します。

- *nnnn* は鍵名です。
- *aaa* はアルゴリズムの数値表現です。
- *iiiii* は鍵識別子 (またはフットプリント) です。

dnssec-keyfromlabel は、表示された文字列に基づいた名前を持つ 2 つのファイルを作成します。K*nnnn*.+*aaa*+*iiiii*.key には公開鍵が、K*nnnn*.+ *aaa*+*iiiii*.private には秘密鍵が含まれます。

最初のファイルには、ゾーンファイルに (直接、または $INCLUDE 文を使用して) 挿入できる DNS KEY レコードが含まれます。

2 番目のファイルには、アルゴリズムに固有のフィールドが含まれます。セキュリティー上の理由から、このファイルには一般的な読み取り権はありません。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/bind
インタフェースの安定性	「流動的」

関連項目

dnssec-keygen(1M), dnssec-signzone(1M), attributes(5)

RFC 2539、RFC 2845、RFC 4033

『BIND 9 Administrator's Reference Manual』を参照してください。このマニュアルページの発行日付時点で、このドキュメントは https://www.isc.org/software/bind/documentation から利用できます。

名前

dnssec-keygen – DNSSEC 鍵生成ツール

形式

```
dnssec-keygen -a algorithm -b keysize -n nametype [-ehk]
      [-c class] [-f flag] [-g generator] [-p protocol]
      [-r randomdev] [-s strength] [-t type] [-v level] name
```

機能説明

dnssec-keygen ユーティリティーは、RFC 2535 および RFC 4034 で定義されている DNSSEC (Secure DNS) の鍵を生成します。また、RFC 2845 で定義されている TSIG (Transaction Signatures) で使用する鍵も生成します。

オプション

次のオプションがサポートされています。

-a *algorithm*
: 暗号化アルゴリズムを選択します。algorithm の値は、RSAMD5 (RSA) または RSASHA1、DSA、NSEC3RSASHA1、NSEC3DSA、DH (Diffie-Hellman)、HMAC-MD5 のいずれかにする必要があります。これらの値は大文字と小文字が区別されません。

  DNSSEC の場合、RSASHA1 は実装が必須のアルゴリズムであるため、DSA が推奨されます。TSIG の場合は、HMAC-MD5 が必須です。

  注 – HMAC-MD5 と DH では、-k フラグが自動的に設定されます。

-b *keysize*
: 鍵のビット数を指定します。鍵サイズの選択は、使用されるアルゴリズムによって異なります。RSAMD5 鍵と RSASHA1 鍵は、512 - 2048 ビットの間にする必要があります。Diffie-Hellman 鍵は、128 - 4096 ビットの間にする必要があります。DSA 鍵は、512 - 1024 ビットの間で、ちょうど 64 の倍数になる値にする必要があります。HMAC-MD5 鍵は、1 - 512 ビットの間にする必要があります。

-c *class*
: 鍵を含む DNS レコードに、指定されたクラスが存在するべきであることを示します。指定されていない場合は、クラス IN が使用されます。

-e
: RSAMD5 鍵または RSASHA1 鍵を生成する場合は、大きな指数を使用します。

-f *flag*
: 指定されたフラグを KEY/DNSKEY レコードのフラグフィールドに設定します。認識されるフラグは KSK (Key Signing Key) DNSKEY のみです。

-g *generator*
: Diffie Hellman 鍵を生成する場合は、この *generator* を使用します。指定可能な値は 2 と 5 です。ジェネレータが指定されていない場合は、可能であれば RFC 2539 からの既知の素数が使用されます。それ以外の場合、デフォルトは 2 です。

-h
: dnssec-keygen のオプションと引数の簡単な要約を出力します。

-k
: DNSKEY レコードではなく KEY レコードを生成します。

-n *nametype*
鍵の所有者型を指定します。*nametype* の値は、ZONE (DNSSEC ゾーン鍵 (KEY/DNSKEY) の場合)、HOST または ENTITY (ホストに関連付けられた鍵 (KEY) の場合)、USER (ユーザーに関連付けられた鍵 (KEY) の場合)、OTHER (DNSKEY) のいずれかにする必要があります。これらの値は大文字と小文字が区別されません。デフォルトは、DNSKEY 生成用の ZONE です。

-p *protocol*
生成された鍵のプロトコル値を設定します。*protocol* 引数は 0 - 255 の数値です。デフォルトは 3 (DNSSEC) です。この引数に指定可能なのほか値は、RFC 2535 およびそれ以降のバージョンに記載されています。

-r *randomdev*
乱数発生元を指定します。オペレーティングシステムによって /dev/random または同等のデバイスが提供されていない場合、デフォルトの乱数発生元はキーボード入力です。*randomdev* は、このデフォルトの代わりに使用される、ランダムデータを含む文字デバイスまたはファイルの名前を指定します。特殊な値「keyboard」は、キーボード入力を使用する必要があることを示します。

-s *strength*
鍵の強さの値を指定します。*strength* 引数は 0 - 15 の数値で、現時点では DNSSEC でその用途は定義されていません。

-t *type*
鍵の使用を示します。type は、AUTHCONF、NOAUTHCONF、NOAUTH、NOCONF のいずれかにする必要があります。デフォルトは AUTHCONF です。AUTH はデータを認証する機能を、CONF はデータを暗号化する機能を示します。

-v *level*
デバッグのレベルを設定します。

生成される鍵

dnssec-keygen が正常に完了すると、K*nnnn*.+*aaa*+*iiiii* の形式の文字列が標準出力に表示されます。これは、生成された鍵の識別文字列です。

- *nnnn* は鍵名です。
- *aaa* はアルゴリズムの数値表現です。
- *iiiii* は鍵識別子 (またはフットプリント) です。

dnssec-keygen ユーティリティーは、出力された文字列に基づいた名前を持つ 2 つのファイルを作成します。

- K*nnnn*.+*aaa*+*iiiii*.key には公開鍵が含まれます。
- K*nnnn*.+*aaa*+*iiiii*.private には秘密鍵が含まれます。

.key ファイルには、ゾーンファイルに (直接、または $INCLUDE 文を使用して) 挿入できる DNS KEY レコードが含まれます。

.private ファイルには、アルゴリズムに固有のフィールドが含まれます。セキュリティー上の理由から、このファイルには一般的な読み取り権はありません。

公開鍵と秘密鍵が同じである場合でも、HMAC-MD5 などの対称暗号化アルゴリズム用に .key ファイルと .private ファイルの両方が生成されます。

使用例

例 1　768 ビットの DSA 鍵の生成

ドメイン example.com 用に 768 ビットの DSA 鍵を生成するには、次のコマンドを発行します。

```
dnssec-keygen -a DSA -b 768 -n ZONE example.com
```

このコマンドは、次の形式の文字列を出力します。

```
Kexample.com.+003+26160
```

次のファイルが作成されます。

```
Kexample.com.+003+26160.key
Kexample.com.+003+26160.private
```

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/bind
インタフェースの安定性	「流動的」

関連項目

dnssec-signzone(1M), attributes(5)

RFC 2539、RFC 2845、RFC 4033

『BIND 9 Administrator's Reference Manual』を参照してください。このマニュアルページの発行日付時点で、このドキュメントは https://www.isc.org/software/bind/documentation から利用できます。

名前

dnssec-makekeyset – DNSSEC zone signing tool

形式

```
dnssec-makekeyset [-ahp] [-s start-time] [-e end-time]
     [-r randomdev] [-t ttl] [-v level] key...
```

機能説明

The `dnssec-makekeyset` utility generates a key set from one or more keys created by `dnssec-keygen(1M)`. It creates a file containing a `KEY` record for each key, and self-signs the key set with each zone key. The output file is of the form *keyset-nnnn.*, where *nnnn* is the zone name.

オプション

`-a` Verify all generated signatures.

`-e` *end-time* Specify the date and time when the generated SIG records expire. As with *start-time*, an absolute time is indicated in `YYYYMMDDHHMMSS` notation. A time relative to the start time is indicated with +*N*, which is *N* seconds from the start time. A time relative to the current time is indicated with now+*N*. If no *end-time* is specified, 30 days from the start time is used as a default.

`-h` Print a short summary of the options and arguments to `dnssec-makekeyset()`.

`-p` Use pseudo-random data when signing the zone. This is faster, but less secure, than using real random data. This option may be useful when signing large zones or when the entropy source is limited.

`-r` *randomdev* Specify the source of randomness. If the operating system does not provide a `/dev/random` or equivalent device, the default source of randomness is keyboard input. The *randomdev* argument specifies the name of a character device or file containing random data to be used instead of the default. The special value `keyboard` indicates that keyboard input should be used.

`-s` *start-time* Specify the date and time when the generated `SIG` records become valid. This can be either an absolute or relative time. An absolute start time is indicated by a number in `YYYYMMDDHHMMSS` notation; 20000530144500 denotes 14:45:00 UTC on May 30th, 2000. A relative start time is indicated by +N, which is N seconds from the current time. If no `start-time` is specified, the current time is used.

`-t` *ttl* Specify the TTL (time to live) of the `KEY` and `SIG` records. The default is 3600 seconds.

`-v` *level* Set the debugging level.

オペランド

The following operands are supported:

key The list of keys to be included in the keyset file. These keys are expressed in the form K*nnnn*.+*aaa*+*iiiii* as generated by `dnssec-keygen`.

使用例

例 1 Generates a keyset containing the DSA key for example.com.

The following command generates a keyset containing the DSA key for `example.com` generated in the `dnssec-keygen(1M)` manual page.

```
dnssec-makekeyset -t 86400 -s 20000701120000 -e +2592000 \
Kexample.com.+003+26160
```

In this example, `dnssec-makekeyset()` creates the file `keyset-example.com`. This file contains the specified key and a self-generated signature.

The DNS administrator for `example.com` could send `keyset-example.com.` to the DNS administrator for `.com` for signing, if the .com zone is DNSSEC-aware and the administrators of the two zones have some mechanism for authenticating each other and exchanging the keys and signatures securely.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWbind9
Interface Stability	Volatile

関連項目

`dnssec-keygen(1M)`, `dnssec-signkey(1M)`, `attributes(5)`

RFC 2535

BIND 9 Administrator Reference Manual

注意事項

Source for BIND9 is available in the SUNWbind9S package.

名前

dnssec-signkey – DNSSEC key set signing tool

形式

```
dnssec-signkey [-ahp] [-c class] [-e end-time]
    [-r randomdev] [-s start-time] [-v level] keyset key...
```

機能説明

The `dnssec-signkey` utility signs a keyset. Typically the keyset will be for a child zone and will have been generated by `dnssec-makekeyset(1M)`. The child zone's keyset is signed with the zone keys for its parent zone. The output file is of the form `signedkey-`*nnnn*., where *nnnn* is the zone name.

オプション

The following options are supported:

`-a` Verify all generated signatures.

`-c` *class* Specify the DNS class of the key sets.

`-e` *end-time* Specify the date and time when the generated SIG records expire. As with *start-time*, an absolute time is indicated in `YYYYMMDDHHMMSS` notation. A time relative to the start time is indicated with +*N*, which is *N* seconds from the start time. A time relative to the current time is indicated with now+*N*. If no *end-time* is specified, 30 days from the start time is used as a default.

`-h` Prints a short summary of the options and arguments to `dnssec-signkey()`.

`-p` Use pseudo-random data when signing the zone. This is faster, but less secure, than using real random data. This option may be useful when signing large zones or when the entropy source is limited.

`-r` *randomdev* Specify the source of randomness. If the operating system does not provide a `/dev/random` or equivalent device, the default source of randomness is keyboard input. *randomdev* specifies the name of a character device or file containing random data to be used instead of the default. The special value `keyboard` indicates that keyboard input should be used.

`-s` *start-time* Specify the date and time when the generated SIG records become valid. This can be either an absolute or relative time. An absolute start time is indicated by a number in `YYYYMMDDHHMMSS` notation; 20000530144500 denotes 14:45:00 UTC on May 30th, 2000. A relative start time is indicated by +*N*, which is *N* seconds from the current time. If no *start-time* is specified, the current time is used.

`-v` *level* Set the debugging level.

オペランド

The following operands are supported:

key The keys used to sign the child's keyset.

keyset The file containing the child's keyset.

使用例

例 1 Sign the keyset file for example.com.

The DNS administrator for a DNSSEC-aware `.com` zone would use the following command to sign the *keyset* file for `example.com` created by `dnssec-makekeyset` with a key generated by `dnssec-keygen`:

```
dnssec-signkey keyset-example.com. Kcom.+003+51944
```

In this example, `dnssec-signkey` creates the file `signedkey-example.com`, which contains the `example.com` keys and the signatures by the `.com` keys.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWbind9
Interface Stability	Volatile

関連項目

`dnssec-keygen(1M)`, `dnssec-makekeyset(1M)`, `dnssec-signzone(1M)`, `attributes(5)`

注意事項

Source for BIND9 is available in the SUNWbind9S package.

名前

dnssec-signzone – DNSSEC ゾーン署名ツール

形式

```
dnssec-signzone [-Aaghptz] [-c class] [-d directory]
    [-e end-time] [-f output-file] [-H iterations] [-I input_format]
    [-i interval] [-k key] [-l domain] [-N soa-serial-format] [-n ncpus]
    [-O output_format] [-o origin] [-r randomdev] [-s start-time]
    [-v level] [-3 salt] zonefile [key]...
```

機能説明

`dnssec-signzone` ユーティリティーはゾーンに署名します。このユーティリティーは `NSEC` レコードと `RRSIG` レコードを生成し、ゾーンの署名されたバージョンを生成します。署名されたゾーンからの委譲のセキュリティーステータス (つまり、子ゾーンがセキュリティー保護されているかどうか) は、子ゾーンごとに `keyset` ファイルが存在するかどうかによって決まります。

オプション

次のオプションがサポートされています。

`-A`
NSEC3 チェーンを生成するときに、すべての NSEC3 レコードに `OPTOUT` フラグを設定し、安全でない委譲に対しては NSEC3 レコードを生成しません。

`-a`
生成されたすべての署名を確認します。

`-c` *class*
ゾーンの `DNS` クラスを指定します。

`-d` *directory*
directory 内の `keyset` ファイルを検索します。

`-e` *end-time*
生成された `RRSIG` レコードの有効期限が切れる日時を指定します。`start-time` と同様に、`YYYYMMDDHHMMSS` の表記で絶対時間が示されます。開始時刻からの相対時間は +*N* で示されます。これは、開始時刻から *N* 秒後であることを示します。現在の時刻からの相対時間は `now+`*N* で示されます。*end-time* が指定されていない場合、開始時刻から 30 日後の日時がデフォルトとして使用されます。

`-f` *output-file*
署名されたゾーンを含む出力ファイルの名前。デフォルトでは、入力ファイル名に `.signed` が付加されます。

`-g`
`keyset` ファイルから子ゾーンの DS レコードを生成します。既存の DS レコードは削除されます。

`-H` *iterations*
NSEC3 チェーンを生成するときに、*iterations* で指定された繰り返しの数を使用します。デフォルトは 100 です。

`-h`
`dnssec-signzone()` のオプションと引数の簡単な要約を出力します。

-I *input-format*
入力ゾーンファイルの形式。指定可能な形式は text (デフォルト) と raw です。このオプションは、動的に署名されたゾーンを主に想定したもので、更新を含む非テキスト形式のダンプされたゾーンファイルに直接署名できるようにします。動的でないゾーンに対してこのオプションを使用しても、意味がありません。

-i *interval*
サイクル間隔 (秒単位) を現在の時刻からのオフセットとして指定します。以前に署名されたゾーンが入力として渡された場合は、レコードに再度署名できます。サイクル間隔のあとで RRSIG レコードの有効期限が切れた場合、そのレコードは保持されます。それ以外の場合は、有効期限が間もなく切れるとみなされ、レコードは置き換えられます。

デフォルトのサイクル間隔は、署名の終了時刻と開始時刻の差の 4 分の 1 です。*end-time* と *start-time* のどちらも指定されていない場合、dnssec-signzone は、有効期間が 30 日でサイクル間隔が 7.5 日の署名を生成します。7.5 日よりも短い期間で有効期限が切れる既存の RRSIG レコードは、すべて置き換えられます。

-j *jitter*
固定された署名有効期間でゾーンに署名すると、署名した時点で発行されたすべての RRSIG レコードが同時に有効期限切れになります。ゾーンが増分的に署名される、つまり、以前に署名されたゾーンが署名者に入力として渡される場合は、期限切れとなるすべての署名をほぼ同じ時刻に再生成する必要があります。jitter オプションは、署名の期限切れ時刻をランダム化するために使用されるジッター時間を指定することにより、増分的な署名の再生成を徐々に分散させます。

署名の有効期間のジッターは、キャッシュの有効期限を分散させるため、バリデータとサーバーにもある程度のメリットをもたらします。つまり、すべてのキャッシュにある多数の RRSIG が同時に有効期限切れにならなければ、すべてのバリデータでほぼ同じ時刻に再取得が必要になる場合に比べて輻輳は少なくなります。

-k *key*
指定された *key* を鍵署名鍵として扱い、鍵フラグをすべて無視します。このオプションは複数回指定できます。

-l *domain*
鍵 (DNSKEY) と DS セットに加えて DLV セットを生成します。ドメインがレコードの名前に付加されます。

-N *soa-serial-format*
署名されたゾーンの SOA シリアル番号形式。指定可能な形式は、次に説明するように keep (デフォルト)、increment、および unixtime です。

keep
SOA シリアル番号を変更しません。

`increment`
RFC 1982 の算術式を使用して SOA シリアル番号を増分します。

`unixtime`
SOA シリアル番号を epoch からの経過秒数に設定します。

`-n` *nthreads*
使用するスレッドの数を指定します。デフォルトでは、検出された CPU ごとに 1 つのスレッドが開始されます。

`-O` *output_format*
署名されたゾーンを含む出力ファイルの形式。指定可能な形式は `text` (デフォルト) と `raw` です。

`-o` *origin*
ゾーンの起点を指定します。指定されていない場合、ゾーンファイルの名前が起点とみなされます。

`-p`
ゾーンに署名するときに疑似乱数データを使用します。この方法は実際のランダムデータを使用する場合に比べて高速ですが、安全性は低下します。このオプションは、大規模なゾーンに署名する場合や、エントロピソースが制限されている場合に役立つことがあります。

`-r` *randomdev*
乱数発生元を指定します。オペレーティングシステムによって `/dev/random` または同等のデバイスが提供されていない場合、デフォルトの乱数発生元はキーボード入力です。*randomdev* は、このデフォルトの `/dev/random` の代わりに使用される、ランダムデータを含む文字デバイスまたはファイルの名前を指定します。特殊な値 `keyboard` は、キーボード入力を使用する必要があることを示します。

`-s` *start-time*
生成された `RRSIG` レコードが有効になる日時を指定します。これは絶対時間または相対時間のどちらでもかまいません。絶対開始時刻は、*YYYYMMDDHHMMSS* という表記の数値で示されます。20000530144500 は、2000 年 5 月 30 日の 14:45:00 UTC のことです。相対開始時刻は +*N* で示されます。これは、現在の時刻から *N* 秒後であることを示します。*start-time* が指定されていない場合は、(クロックスキューを考慮して) 現在の時刻から 1 時間前の時刻が使用されます。

`-t`
完了時に統計情報を出力します。

`-v` *level*
デバッグのレベルを設定します。

`-z`
署名する対象を決定するときに、鍵の KSK フラグを無視します。

-3 *salt*
指定された 16 進数でエンコードされた *salt* を持つ NSEC3 チェーンを生成します。ダッシュ (-) を使用すると、NSEC3 チェーンを生成するときにソルトが使用されないことを示すことができます。

オペランド

次のオペランドがサポートされています。

zonefile
署名されるゾーンを含むファイル。

key
ゾーンに署名するためにどの鍵を使用するべきかを指定します。鍵が指定されていない場合は、ゾーンの頂点に DNSKEY レコードがないかどうかが検証されます。レコードが見つかり、かつ一致する秘密鍵が現在のディレクトリ内に存在する場合は、これらの秘密鍵が署名に使用されます。

使用例

例 1　DSA 鍵によるゾーンへの署名

次のコマンドは、dnssec-keygen(1M) のマニュアルページにある例で生成された DSA 鍵を使用して example.com ゾーンに署名します (Kexample.com.+003+17247)。ゾーンの鍵がマスターファイル (db.example.com) に存在する必要があります。この呼び出しでは、現在のディレクトリ内で鍵セットファイルを検索することにより、それらのファイルから DS レコードを生成できるようにしています (-g)。

```
% dnssec-signzone -g -o example.com db.example.com \
Kexample.com.+003+17247
db.example.com.signed
%
```

上の例では、dnssec-signzone はファイル db.example.com.signed を作成します。このファイルは、named.conf ファイル内の zone 文で参照される必要があります。

例 2　以前に署名されたゾーンへの再署名

次のコマンドは、デフォルトのパラメータを使用して、以前に署名されたゾーンに再署名します。秘密鍵が現在のディレクトリに存在しているものとします。

```
% cp db.example.com.signed db.example.com
% dnssec-signzone -o example.com db.example.com \
db.example.com.signed
%
```

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/bind

属性タイプ	属性値
インタフェースの安定性	「流動的」

関連項目

dnssec-keygen(1M), attributes(5)

RFC 4033

『BIND 9 Administrator's Reference Manual』を参照してください。このマニュアルページの発行日付時点で、このドキュメントは https://www.isc.org/software/bind/documentation から利用できます。

名前

domainname – set or display name of the current domain

形式

`domainname [name-of-domain]`

機能説明

Without an argument, `domainname` displays the name of the current domain name used in RPC exchanges, usually referred to as the NIS domain name. This name typically encompasses a group of `hosts` or `passwd` entries under the same administration. The `domainname` command is used by various components of Solaris to resolve names for entries such as are found in `passwd`, `hosts` and `aliases`. By default, naming services such as NIS use `domainname` to resolve names.

With `solaris.smf.manage.domainname` rbac(5) privileges, you can set the name of the domain by specifying the domain name as the value of the `config/domainname` property of the `nis/domain` service with a sequence of commands such as the following:

```
% svccfg -s nis/domain setprop config/domainname = \
hostname: nisdomain.example.com
% svccfg -s nis/domain:default refresh
% svcadm enable nis/domain
```

In addition to setting the value of `config/domainname`, the preceding sequence enables the `nis/domain` service. This service is not enabled by default. The `nis/domain` service sets up all the shared NIS configuration, of which `domainname` is the one mandatory configuration parameter.

The domain name for various naming services can also be set by other means. For example, `ypinit` can be used to specify a different domain for all NIS calls. The domain name of the machine is usually set during boot time through the `domainname` command by the `svc:/system/identity:domain` service. If the new domain name is not saved in the `/etc/defaultdomain` file, the machine reverts to the old domain after it reboots.

The sendmail(1M) daemon, as shipped with Solaris, and the `sendmail` implementation provided by `sendmail.org` (formerly referred to as “Berkeley 8.x sendmail”) both attempt to determine a local host's fully qualified host name at startup and both pursue follow-up actions if the initial search fails. It is in these follow-up actions that the two implementations differ.

Both implementations use a standard Solaris or Unix system call to determine its fully qualified host name at startup, following the name service priorities specified in nsswitch.conf(4). To this point, the Solaris and `sendmail.org` versions behave identically.

If the request for a fully qualified host name fails, the `sendmail.org` `sendmail` sleeps for 60 seconds, tries again, and, upon continuing failure, resorts to a short name. The Solaris version of `sendmail` makes the same initial request, but then, following initial failure, calls `domainname`. If successful, the sleep is avoided.

On a Solaris machine, if you run the `sendmail.org` version of sendmail, you get the startup behavior (omitting the `domainname` call) described above. If you run the Solaris `sendmail`, the `domainname` call is made if needed.

If the Solaris sendmail cannot determine the fully qualified host name, use check-hostname(1M) as a troubleshooting aid. This script can offer guidance as to appropriate corrective action.

ファイル

- /etc/defaultdomain
- /etc/nsswitch.conf

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

svcs(1), check-hostname(1M), hostconfig(1M), named(1M), sendmail(1M), svcadm(1M), svccfg(1M), ypinit(1M), aliases(4), defaultdomain(4), hosts(4), nsswitch.conf(4), passwd(4), attributes(5), rbac(5), smf(5)

注意事項

The domainname service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/identity:domain
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

drd – Logical Domain Dynamic Reconfiguration daemon

形式

`/usr/lib/ldoms/drd`

機能説明

The `drd` daemon is part of the framework that enables the addition and removal of resources from a Logical Domain. This framework is collectively called Dynamic Reconfiguration (DR).

`drd` is responsible for various aspects of DR on a Logical Domain and must be enabled to ensure proper DR functionality. It is started at boot time and has no configuration options.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/ldoms
Interface Stability	Uncommitted

関連項目

`svcs(1)`, `svcadm(1M)`, `syslog(3C)`, `syslog.conf(4)`, `attributes(5)`, `smf(5)`

エラー

`drd` uses `syslog(3C)` to report status and error messages. All of the messages are logged with the `LOG_DAEMON` facility. Error messages are logged with the `LOG_ERR` and `LOG_NOTICE` priorities, and informational messages are logged with the `LOG_INFO` priority. The default entries in the `/etc/syslog.conf` file log all the `drd` error messages to the `/var/adm/messages` log.

注意事項

The `drd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/platform/sun4v/drd:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

drvconfig – apply permission and ownership changes to devices

形式

```
drvconfig [-bn] [-a alias_name] [-c class_name]
     [-i drivername] [-m major_num] [-r root_dir]
```

機能説明

devfsadm(1M) is now the preferred command and should be used instead of drvconfig.

The default operation of drvconfig is to apply permission and ownership changes to devices. Normally, this command is run automatically after a new driver has been installed (with add_drv(1M)) and the system has been rebooted.

オプション

The following options are supported:

-a*alias_name*	Add the name *alias_name* to the list of aliases that this driver is known by. This option, if used, must be used with the -m *major_num*, the -b and the -i *drivername* options.
-b	Add a new major number to name binding into the kernel's internal name_to_major tables. This option is not normally used directly, but is used by other utilities such as add_drv(1M). Use of the -b option requires that -i and -m be used also. No /devices entries are created.
-c*class_name*	The driver being added to the system exports the class *class_name*. This option is not normally used directly, but is used by other utilities. It is only effective when used with the -b option.
-i*drivername*	Only configure the devices for the named driver. The following options are used by the implementation of add_drv(1M) and rem_drv(1M), and may not be supported in future versions of Solaris:
-m*major_num*	Specify the major number *major_num* for this driver to add to the kernel's name_to_major binding tables.
-n	Do not try to load and attach any drivers, or if the -i option is given, do not try to attach the driver named *drivername*.
-r*root_dir*	Perform operations under *root_dir*, rather than directly under root.

終了ステータス

0	Successful completion.
non-zero	An error occurred.

ファイル

/devices	Device nodes directory
/etc/minor_perm	Minor mode permissions
/etc/name_to_major	Major number binding
/etc/driver_classes	Driver class binding file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

sh(1), add_drv(1M), modinfo(1M), modload(1M), modunload(1M), rem_drv(1M), update_drv(1M), path_to_inst(4), attributes(5), devfs(7FS)

名前

dsbitmap – Availability Suite ビットマップボリュームのサイズ計算

形式

```
dsbitmap -h
dsbitmap -p data_volume [bitmap_volume]
dsbitmap -r data_volume [bitmap_volume]
```

機能説明

`dsbitmap` コマンドは、指定されたデータボリュームで使用するために必要な Availability Suite ビットマップボリュームのサイズを計算します。

オプション

次のオプションがサポートされています。

`-h`
: `dsbitmap` コマンドの使用法に関するメッセージを出力します。

`-p` *data_volume* [*bitmap_volume*]
: 指定された *data_volume* について、`dsbitmap` は、関連付けられている Availability Suite ポイントインタイムのビットマップボリュームにとって必要なサイズを計算して表示します。可能性のあるすべての Availability Suite ポイントインタイムセット構成のビットマップボリュームサイズが表示されます。これらの構成には、独立シャドウ、フルサイズ依存シャドウ、および小型依存シャドウが含まれます。オプションの *bitmap_volume* 引数が指定された場合、`dsbitmap` は、このボリュームが *data_volume* のビットマップボリュームとして使用するための十分な大きさがあるかどうかを判定します。

`-r` *data_volume* [*bitmap_volume*]
: 指定された *data_volume* について、`dsbitmap` は、関連付けられている Availability Suite 遠隔ミラーのビットマップボリュームにとって必要なサイズを計算して表示します。可能性のあるすべての Availability Suite 遠隔ミラーセット構成のビットマップボリュームサイズが表示されます。これらの構成には、同期複製のほか、メモリーキュー、ディスクキュー、および 32 ビットの refcount を備えた非同期複製が含まれます。オプションの *bitmap_volume* 引数が指定された場合、`dsbitmap` は、このボリュームが *data_volume* のビットマップボリュームとして使用するための十分な大きさがあるかどうかを判定します。

使用法

`dsbitmap` は通常、必要なビットマップボリュームサイズを判定したあと、作成されたビットマップボリュームが適切かどうかを確認するために、Sun StorageTek Availability Suite ソフトウェアの初期の構成段階でシステム管理者によって使用されます。

終了ステータス

次の終了値が返されます。

`0`
: 正常終了。ビットマップボリュームの名前が指定された場合は、そのボリュームのサイズが十分な大きさで、可能性のあるすべての使用方法に対応できることを示します。

`1`
: エラーが発生しました。

`2`
: コマンド行で無効なオプションが指定されました。

3 指定されたビットマップボリュームは、ディスクキューを備えた非同期セットの Availability Suite 遠隔ミラービットマップとして使用するための十分な大きさはありませんが、ほかのすべての遠隔ミラーセット構成に使用するための十分な大きさはあります。

4 指定されたビットマップボリュームは、遠隔ミラーセット構成の Availability Suite 遠隔ミラービットマップとして使用するための十分な大きさがありません。

5 指定されたビットマップボリュームは、遠隔ミラーセット構成の Availability Suite 遠隔ミラービットマップとして使用するための十分な大きさがありません。

6 指定されたビットマップボリュームは、小型依存シャドウの Availability Suite ポイントインタイムビットマップとして使用するための十分な大きさがありませんが、ほかのすべてのポイントインタイムセット構成に使用できるだけの大きさは持っています。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目

iiadm(1M), sndradm(1M), attributes(5)

名前

dscfg – Sun StorageTek Availability Suite ソフトウェアの構成ツール

形式

```
dscfg
dscfg -L
dscfg -h
dscfg [-C group]
dscfg [-C group] -i [-p parser_config_file]
dscfg [-C group] -a config_file
dscfg [-C group] [-l]
dscfg [-C group] [-l] -s config_location
dscfg -D dgname
```

機能説明

`dscfg` コマンドは、構成データベースを初期化、一覧表示、フォーマット、および復元する機能を提供することによって Availability Suite 構成を制御します。

オプション

オプションが指定されていない場合、`dscfg` は現在のローカル構成の場所を表示します。`dscfg` コマンドは、次のオプションをサポートします。

`-L`
構成データベースへのアクセスを制御するロックのステータスを表示します。構成データベースがロックされている場合は、ロックのタイプ (読み取りまたは書き込み) が、そのロックを保持するプロセスのプロセス ID とともに表示されます。

`-h`
`dscfg` コマンドの使用法に関するメッセージを表示します。

`-i`
構成データベースを初期化します。以前または現在の構成情報が削除されるため、このオプションでは、続行する前に削除を確認するプロンプトが表示されます。

`-p` *parser_config_file*
`-i` オプションとともに使用された場合、`dscfg` はパーサー構成ファイルを永続構成に読み込み、構成データベースを再フォーマットします。

`-a` *config_file*
指定された *config_file* を構成に復元します。このオプションでは、ファイルのエラーチェックは行われません。このオプションを使用すると、構成ファイルが無効になります。

`-l`
現在の構成データベースの内容を `-a` オプションに適した形式で一覧表示します。`-s` オプションと組み合わせて使用された場合は、`-s` オプションに渡された場所に格納されている内容を表示しますが、構成の場所は設定しません。

次のオプションは、Sun Cluster で構成されたシステムでのみ有効です。*group* には、Sun Cluster デバイスグループ、またはローカルデバイスを示す '-' のどちらかを指定できます。

`-C` *group*	クラスタ構成データベースの場所を表示します。
`-C` *group* `-i`	指定されたグループにのみ関連付けられた構成データベースエントリを初期化します。以前または現在の構成情報が削除されるため、このオプションでは、続行する前に削除を確認するプロンプトが表示されます。
`-C` *group* `-p` *parser_config_file*	`-i` オプションとともに使用された場合、`dscfg` はパーサー構成ファイルを永続的な構成に読み込み、指定されたグループにのみ関連付けられた構成データベースエントリを再フォーマットします。
`-C` *group* `-a` *config_file*	指定された *config_file* を、指定されたグループにのみ関連付けられた構成データベースエントリに復元します。このオプションでは、ファイルのエラーチェックは行われません。このオプションを使用すると、構成ファイルが無効になります。
`-C` *group* `-l`	指定されたリソースグループに関連付けられた現在の構成データベースの内容を一覧表示します。
`-C` *group* `-s` *config_file_location*	Sun Cluster 構成データベースが存在する場所を指定します。この場所は、DID デバイスである必要があります。
`-D` *device_group*	指定された *device_group* が Sun Cluster によって認識されているかどうか、およびこのノード上で使用可能かどうかを確認します。デバイスグループのステータスを示す情報を表示します。戻り値は次のとおりです。 (echo $? を使用して戻り値を取得します) `1` デバイスグループは Sun Cluster 内にあり、このノード上でアクティブです。 `0` デバイスグループは Sun Cluster 内にありますが、別のノード上でアクティブです。 `–1` デバイスグループは Sun Cluster 内にありません。

使用法

dscfg コマンドは通常、パッケージインストールスクリプトなどのほかのスクリプトや、dscfgadm(1M) から実行されます。これは一般的な用途向けではありません。dscfgadm に含まれていない dscfg の使用については、『Availability Suite Troubleshooting Guide』を参照してください。

ファイル

/etc/dscfg_format
: パーサー構成ファイル

/etc/dscfg_local
: クラスタタグを持たないすべてのエントリ用の構成の場所

/etc/dscfg_cluster
: Sun Cluster 構成データベースの場所を指定する参照ファイル

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目

dscfgadm(1M), iiadm(1M), scmadm(1M), sndradm(1M), svadm(1M), ds.log(4), attributes(5)

名前

dscfgadm – Availability Suite 構成の場所およびサービスのための管理コマンド

形式

```
dscfgadm -i

dscfgadm -e [-r] [-p]

dscfgadm -d [-r]

dscfgadm -s
```

機能説明

dscfgadm コマンドは、構成の場所を設定したり、Availability Suite サービスを有効または無効にしたりする機能を提供することによって、Availability Suite 構成の場所とサービスを制御します。

オプション

dscfgadm コマンドの引数を指定しない場合、このユーティリティーでは、構成の場所の設定やサービスの開始を対話形式で実行できます。構成の場所は、サイズやファイルタイプなどの条件を満たしていることを確認するために検証されてから、使用のために初期化されます。

dscfgadm コマンドは、次のオプションをサポートします。

-d [-r]
: Availability Suite SMF サービスを無効にします。これには、デーモンの停止や、現在 Availability Suite ソフトウェアの下で有効になっているすべてのポイントインタイムコピーまたは遠隔ミラーセットの中断が含まれます。追加オプションなしで実行された場合は、-d オプションによって、すべての Availability Suite サービスが無効になります。この設定はシステムのブート後も永続します。

  -r
  : -d または -e オプションに渡された場合は、遠隔ミラーサービス上でのみ動作します。

-e [-r] [-p]
: Availability Suite SMF サービスを有効にします。これには、デーモンの起動や、以前に Availability Suite ソフトウェアの下で構成されたすべてのポイントインタイムコピーまたは遠隔ミラーセットの再開が含まれます。追加オプションなしで実行された場合は、-e オプションによって、すべての Availability Suite サービスが有効になります。この設定はシステムのブート後も永続します。

  -r
  : -d または -e オプションに渡された場合は、遠隔ミラーサービス上でのみ動作します。

  -p
  : -e オプションに渡された場合は、ポイントインタイムコピーサービスのみが有効になります。

-i
: Availability Suite SMF サービスに関する情報を表示します。

`-s`
Sun Cluster 構成に固有の情報を含む構成データベースの場所を設定できるようにします。

`-x`
プログラム実行中に詳細なデバッグ情報を表示します。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目

`dscfg(1M)`, `iiadm(1M)`, `scmadm(1M)`, `sndradm(1M)`, `svadm(1M)`, `attributes(5)`

名前

dscfglockd – Availability Suite サービスのロックデーモン

形式

`/usr/lib/dscfglockd [-e program | -f file]`

機能説明

`dscfglockd` デーモンは、クラスタ内のノード間の StorageTek 構成データベースロック要求を調整します。

オプション

`dscfglockd` は、次のオプションをサポートします。

`-e` *program*
: *program* 内のスクリプトと引数を実行します。実行可能ファイルとそれに必要なすべての引数は、このオプションに対する単一の引数として、適切な引用で指定される必要があります。この引数は `sh(1)` によって処理されます。

`-f` *file*
: *file* からピアホストの名前のリストを読み取ります。

引数が指定されていない場合、`dscfglockd` はローカルのロックデーモンとして機能しますが、独自の構成プロセスの一部としてローカルホストに接続する可能性のあるほかのすべてのデーモンとロック要求を調整します。

終了ステータス

`0` デーモンが正常に起動されました。

`>0` デーモンの起動に失敗しました。

ファイル

`/lib/svc/method/svc-scm`
: `dscfglockd` を起動および停止するためのシェルスクリプト。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目

`sh(1)`, `attributes(5)`

名前 dsstat – Sun StorageTek Availability Suite の入出力統計の報告

形式

```
dsstat -m mode [-r report–options] [-d display-options]
 [-s volume-sets] [-f | -F] [-z] [interval [count]]
```

機能説明 dsstat コマンドは、Sun StorageTek Availability Suite 製品の入出力統計を収集して報告します。

オプション dsstat は、次のオプションをサポートします。

-m *mode*

操作のモードを指定します。有効なモードは次のとおりです。

cache
: キャッシュの統計情報

ii
: ポイントインタイムコピーの統計情報

sndr
: 遠隔ミラーの統計情報

複数のサービスの統計情報を同時に表示するには、-m スイッチを使用して、必要な各モードを指定します。例:

```
% dsstat -m ii -m sndr
% dsstat -m ii -m sndr -m cache
```

どのサービスからどの統計情報が報告されているかを判定するには、後述の *role* フィールドを使用します。複数のサービスの 1 つとしてキャッシュの統計情報が要求された場合は、rkps と wkps の統計情報が、crkps、drkps、cwkps、および dwkps にさらに分けられます。-m スイッチが指定されていない場合、dsstat は、前述したすべての有効なモードのデフォルトの統計情報を表示します。これらのフィールドの説明については、後述の「フィールドの説明」を参照してください。

-r *report–options*

表示されるボリュームコンポーネントを指定します。各項目が 1 文字で表され、複数の項目を選択できます。*report–options* は、上で指定されたモードによって変わります。このオプションは、cache モードでは使用されません。

ii モードで有効な *report-options* は次のとおりです。

m
: マスターボリュームの統計情報。

- 「ii モードの report-options」の場合: m、s、b、o
- 「sndr モードの report-options」の場合: b、n
- 「cache モードの display-options」の場合: r、w、d、c、s、f
- 「ii モードの display-options」の場合: r、w、t、s、p、f

s
: シャドウボリュームの統計情報。

b　　ビットマップボリュームの統計情報。

o　　オーバーフローボリュームの統計情報 (接続されている場合)。

sndr モードで有効な *report–options* は次のとおりです。

b　　ビットマップボリュームの統計情報。

n　　ネットワークボリュームの統計情報。

-d *display–options*
表示される統計情報を指定します。統計情報のタイプが 1 文字で表され、複数のタイプを指定できます。

cache モードで有効な *display-options* は次のとおりです。

r　　詳細な読み取りの統計情報。

w　　詳細な書き込みの統計情報。

s　　要約の統計情報。

f　　キャッシュの動作フラグ。

次の *display–options* は cache モードでのみ使用可能で、モードのオプション (-m) と組み合わせる必要があります。

d　　デステージされたデータの統計情報。

c　　書き込み取り消しの統計情報。

cache のデフォルトの *display-options* は sf です。

ii モードで有効な *display-options* は次のとおりです。

r
詳細な読み取りの統計情報。

w
詳細な書き込みの統計情報。

t
タイミングの統計情報。

s
要約の統計情報。

p
同期が必要なボリュームの割合 (パーセント)。

f
ボリュームタイプ/ステータスフラグ。

ii のデフォルトの *display-options* は spf です。sndr モードで有効な *display-options* は次のとおりです。

r　詳細な読み取りの統計情報。

w　詳細な書き込みの統計情報。

t　タイミングの統計情報。

s　要約の統計情報。

p　同期が必要なボリュームの割合 (パーセント)。

f　ボリュームタイプ/ステータスフラグ。

q　非同期待ち行列の統計情報。

次の *display-option* は sndr モードでのみ使用可能で、モードのオプション (-m) と組み合わせる必要があります。

q　非同期待ち行列の統計情報。

sndr のデフォルトの *display-options* は spf です。

要約の *display–option* を指定すると、rwt の *display?options* が無視されます。

-s *volume–sets*　指定された *volume–sets* のみを含むように出力をフィルタ処理します。ここで、*volume-sets* はボリューム名のコンマ区切りリストです。遠隔ミラーの統計情報を表示する場合は、指定された名前を遠隔ミラーの一次ボリュームと比較して、一致するかどうかを判定します。さらに、遠隔ミラーボリュームを <*host*>:<*volume*> として指定する必要があります。ここで、<*host*> は一次ホストまたは二次ホストで、<*volume*> はそのホストがボリュームセットを認識するために使用するボリューム名です。複数の遠隔ミラーボリュームセットを指定する場合は、指定した各ボリュームセットが、上の規則に基づいて個別に評価されることに注意してください。ポイントインタイムコピーの統計情報を表示する場合は、指定されたボリューム名をポイントインタイムコピーのシャドウボリュームと比較して、一致するかどうかを判定します。

-f　すべての報告サイクルでフィールドヘッダーを出力します。

-F　報告が開始されるときに 1 回だけフィールドヘッダーを出力します。

-z　ゼロ値が含まれているか、またはアクティビティーが存在しないレポート行を抑制します。

オペランド

dsstat コマンド行は、次のオペランドをサポートします。

interval
各レポートの *interval* を指定します (秒)。*interval* が指定されていない場合、*interval* が 1 秒のレポートが 1 つ出力されます。

特定のセット内のエントリの数は変動し、セットの数も変動するために、セットが報告されている最中にヘッダー情報が表示されることがあります。これを回避するには、-f または -F オプションを使用して、ヘッダー情報を希望する時間に表示します。

count
生成するレポートの数を指定します。*count* が指定されていない場合は、中断されるまで出力が続行されます。

フィールドの説明

特に指定されていないかぎり、すべてのフィールドが、指定された *interval* 中に収集されたデータに基づく 1 秒あたりの平均になります。

name	報告されているエンティティーの名前
t	ボリュームタイプ
s	ボリュームステータス
r	キャッシュの読み取り動作
w	キャッシュの書き込み動作
pct	同期が必要なボリュームの割合 (パーセント)
role	報告されている項目の役割
tps	読み取り + 書き込みの合計数
rtps	読み取りの数
wtps	書き込みの数
kps	読み取り + 書き込みが行われた合計の K バイト数
rkps	読み取られた K バイト数
wkps	書き込まれた K バイト数
crkps	キャッシュから読み取られた K バイト数
drkps	ディスクから読み取られた K バイト数
cwkps	キャッシュに書き込まれた K バイト数
dwkps	ディスクに書き込まれた K バイト数
ckps	キャッシュとの間で転送された K バイト数
dkps	ディスクとの間で転送された K バイト数

svt	操作ごとのサービス時間
hit	*interval* 中の読み取りヒット数
ds/s	キャッシュからデステージされた K バイト数
cn/s	書き込み取り消しの数
q	使用されている非同期待ち行列のタイプ
qi	現在待ち行列に入っている項目の数
qk	現在待ち行列に入っている K バイト数
qhwi	待ち行列に入っている項目の高位境界値
qhwk	待ち行列に入っている K バイト数の高位境界値

name フィールドには、名前の最後の 16 文字のみが表示されます。

cache で有効なキャッシュ動作は、次のとおりです。

C	キャッシュの読み取り/書き込み
D	ディスクの読み取り/書き込み

ii で有効なボリュームタイプは次のとおりです。

I	独立シャドウボリューム
D	依存シャドウボリューム

ii で有効なボリュームステータスは次のとおりです。

C	コピーが進行中
-	コピーが進行中ではない

sndr で有効なボリュームタイプは次のとおりです。

P	これは、このボリュームセットの一次ホストです。
S	これは、このボリュームセットの二次ホストです。

sndr で有効なボリュームステータスは次のとおりです。

L	このボリュームへの変更がログに記録されています
Q	ボリュームが待ち行列モードにあります
R	変更を二次に複製しています
SY	同期が進行中です。データを送信しています。
RS	逆同期が進行中です。データを受信しています。

SN　　同期が必要です

RN　　逆同期が必要です

VF　　ボリュームが失敗しました

BF　　ビットマップが失敗しました

QF　　待ち行列が失敗しました

sndr で有効な待ち行列タイプは次のとおりです。

D　　ディスクベースの待ち行列が有効になっています。

M　　メモリーベースの待ち行列が有効になっています。

sndr のボリュームの役割は次のとおりです。

net　　ネットワークボリュームの統計情報

bmp　　ビットマップボリュームの統計情報

ii のボリュームの役割は次のとおりです。

mst　　マスターボリュームの統計情報

shd　　シャドウボリュームの統計情報

bmp　　ビットマップボリュームの統計情報

ovr　　オーバーフローボリュームの統計情報

使用例

例 1　キャッシュの統計情報の報告

次の例では、キャッシュの統計情報の報告を、キャッシュ/ディスクへの読み取りと書き込みの詳細な内訳を含めて表示しています。レポートは 5 秒間隔で生成されます。報告される内容はセット /dev/rdsk/c1t1d0s0 に制限されています。

```
 # dsstat -m cache -d rw -s /dev/rdsk/c1t1d0s0 5
- read -             - write -
name                ckps   dkps    hit   ckps   dkps    hit
dev/rdsk/c1t1d0s0      0      0   0.00      0      0   0.00
dev/rdsk/c1t1d0s0      3   2396   0.13    983    763 100.00
dev/rdsk/c1t1d0s0   2399    799  75.00   2815   2686 100.00
dev/rdsk/c1t1d0s0   3200    800  80.00   2755   2908 100.00
dev/rdsk/c1t1d0s0   3999    799  83.33   2809   2868 100.00
dev/rdsk/c1t1d0s0   4800    800  85.71   2867   2931 100.00
```

例 2　マスター、シャドウ、およびビットマップの統計情報の報告

デフォルトの出力を使用して、ポイントインタイムコピーのマスター、シャドウ、およびビットマップの統計情報を報告します。レポートは 2 秒間隔で生成されます。

例 2　マスター、シャドウ、およびビットマップの統計情報の報告　　(続き)

```
# dsstat -m ii -r msb 2
name               t  s    pct role    kps   tps  svt
dev/rdsk/c0t1d0s5  I  C  96.15  mst  19921    38   22
dev/rdsk/c0t1d0s6               shd   9960    19   20
dev/rdsk/c0t1d0s7               bmp     39    77    2
dev/rdsk/c0t1d0s5  I  C  94.24  mst  19623    38   22
dev/rdsk/c0t1d0s6               shd   9939    19   20
dev/rdsk/c0t1d0s7               bmp     39    77    2
dev/rdsk/c0t1d0s5  I  C  92.34  mst  19969    39   22
dev/rdsk/c0t1d0s6               shd   9984    19   20
dev/rdsk/c0t1d0s7               bmp     39    78    2
```

例 3　遠隔ミラーのネットワークの統計情報の報告

詳細な読み取りと書き込みの統計情報を使用して、遠隔ミラーのネットワークの統計情報を報告します。レポートには、ボリュームタイプ/ステータスフラグと割合(パーセント)が含まれます。レポートは 2 秒間隔で生成されます。報告される内容はセット `/dev/rdsk/c0t1d0s0` に制限されています。

```
# dsstat -m sndr -r n -d rwpf -s /dev/rdsk/c0t1d0s0 2
name               t  s    pct role   rkps  rtps   wkps  wtps
dev/rdsk/c0t1d0s0  P  L 100.00  sec      0     0      0     0
dev/rdsk/c0t1d0s0  P SY  99.90  sec      0     0    288     9
dev/rdsk/c0t1d0s0  P SY  97.90  sec      0     0   5296   165
dev/rdsk/c0t1d0s0  P SY  95.81  sec      0     0   5184   161
dev/rdsk/c0t1d0s0  P SY  93.81  sec      0     0   5280   164
dev/rdsk/c0t1d0s0  P SY  91.71  sec      0     0   5198   162
```

終了ステータス

次の終了値が返されます。

`0`　正常終了。

`1`　正常終了。報告する統計情報がありません。

`2`　無効な引数が検出されました。

`3`　`kstat` の統計情報を作成するために使用できるメモリーがありません。

`4`　原因不明のエラーが発生しました。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目 dscfg(1M), svadm(1M), ds.log(4), rdc.cf(4), attributes(5)

名前

dsvclockd – DHCP service lock daemon

形式

```
/usr/lib/inet/dsvclockd [-d 1 | 2] [-f ] [-v]
```

機能説明

The `dsvclockd` daemon is a lock manager that works in conjunction with the Dynamic Host Configuration Protocol (DHCP) Data Service Library (`libdhcpsvc`). It provides shared or exclusive access to the `dhcp_network(4)` and `dhcptab(4)` tables. This service is used by the `SUNWbinfiles` and `SUNWfiles` DHCP data store modules. See `dhcp_modules(5)`.

`dsvclockd` is started on demand by `libdhcpsvc`. The `dsvclockd` daemon should be started manually only if command line options need to be specified.

オプション

The following options are supported:

`-d 1 | 2` Set debug level. Two levels of debugging are currently available, `1` and `2`. Level `2` is more verbose.

`-f` Run in the foreground instead of as a daemon process. When this option is used, messages are sent to standard error instead of to `syslog(3C)`.

`-v` Provide verbose output.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dhcp
Interface Stability	Uncommitted

関連項目

`syslog(3C)`, `dhcp_network(4)`, `dhcptab(4)`, `dhcp_modules(5)`, `attributes(5)`

名前

dtrace – DTrace 動的トレースコンパイラおよびトレースユーティリティー

形式

```
dtrace [-32 | -64] [-aACeFGHhlqSvVwZ] [-b bufsz] [-c cmd]
     [-D name [=value]] [-I path] [-L path] [-o output]
     [-s script] [-U name] [-x arg [=val]]
     [-X a | c | s | t] [-p pid]
     [-P provider [[predicate] action]]
     [-m [provider:] module [[predicate] action]]
     [-f [[provider:] module:] function [[predicate] action]]
     [-n [[[provider:] module:] function:] name [[predicate] action]]
     [-i probe-id [[predicate] action]]
```

機能説明

DTrace は、Solaris オペレーティングシステム用の総合的な動的トレースフレームワークです。DTrace が提供する強力なインフラストラクチャーを使えば、管理者、開発者、およびサービス担当者は、オペレーティングシステムやユーザープログラムの動作に関するさまざまな質問に対して、簡潔に答えることができるようになります。

『『Solaris Dynamic Tracing Guide』』では、DTrace を使用してシステムの動作を監視、デバッグ、およびチューニングする方法について説明します。バンドルされている DTrace の監視ツール、計測プロバイダ、D プログラミング言語など、DTrace の各種機能の詳細については、このマニュアルを参照してください。

`dtrace` コマンドは、DTrace 機能が提供する基本サービスに対する汎用インタフェースを提供します。それらを次に示します。

- DTrace によって現在公開されている一連のプローブやプロバイダを一覧表示するためのオプション
- 任意のプローブ記述指定子 (provider、module、function、name) を使ってプローブを直接有効にするためのオプション
- D コンパイラを実行し、1 つ以上の D プログラムファイルまたはコマンド行に直接記述されたプログラムをコンパイルするためのオプション
- 匿名トレースプログラムを生成するためのオプション
- プログラム安定性レポートを生成するためのオプション
- DTrace のトレース動作やバッファリング動作を変更したり、追加の D コンパイラ機能を有効化したりするためのオプション

`dtrace` を使えば、D スクリプトを作成できます。それには、`#!` 宣言内にこの dtrace を指定してインタプリタファイルを作成します。また、`dtrace` を使えば、実際に `-e` オプションを使用してトレースを有効化することなく D プログラムのコンパイルを試行し、そのプロパティーを決定することもできます。詳しくは「オプション」の節を参照してください。`dtrace` ユーティリティーを使ってこれらのタスクを実行する方法に関する詳しい例については、『『Solaris Dynamic Tracing Guide』』を参照してください。

オプション

-P、-m、-f、-n、および -i オプションが受け入れる引数には、1 つの D 言語 *predicate* をスラッシュ // で囲んだものと、一連の D 言語 *action* 文を中括弧 {} で囲んだものを、オプションで含めることができます。コマンド行で指定される D プログラムのコードは、メタ文字がシェルによって解釈されてしまわないように、適切に引用符で囲む必要があります。

次のオプションがサポートされています。

-32 | -64
: D コンパイラは、オペレーティングシステムカーネルのネイティブデータモデルを使ってプログラムを生成します。isainfo -b コマンドを使えば、現在のオペレーティングシステムのデータモデルを確認できます。-32 オプションが指定された場合、dtrace は、32 ビットデータモデルを使って D プログラムをコンパイルするよう、D コンパイラに指示します。-64 オプションが指定された場合、dtrace は、64 ビットデータモデルを使って D プログラムをコンパイルするよう、D コンパイラに指示します。通常、これらのオプションを指定する必要はありません。オプションを省略しても、dtrace は、自動的にネイティブデータモデルを選択します。データモデルは、整数型のサイズとその他の言語プロパティに影響を及ぼします。いずれかのデータモデル用にコンパイルされた D プログラムは、32 ビットカーネル上でも、64 ビットカーネル上でも実行できます。オプション -32 と -64 では、-G オプションを指定したとき生成される ELF ファイルの形式 (ELF32 または ELF64) も指定できます。

-a
: 匿名トレース状態を要求し、トレースされたデータを表示します。-a オプションと -e オプションを併用した場合、dtrace は、匿名トレース状態の終了後、新しいデータを待たずにただちに終了します。匿名トレースの詳細については、『『Solaris Dynamic Tracing Guide』』を参照してください。

-A
: 匿名トレース用の driver.conf(4) 指令を生成します。このオプションは、指定されたプローブが匿名トレース用に有効化されてから終了するまでの一連の dtrace(7D) 構成ファイル指令を作成します。デフォルトでは、dtrace は、指令を /kernel/drv/dtrace.conf ファイルに格納します。-o オプションを使って別の出力ファイルを指定すれば、この動作を変更できます。

-b *bufsz*
: 主トレースバッファーサイズ (*bufsz*) を設定します。このトレースバッファーサイズには、サイズサフィックス k、m、g、t のいずれかを含めることができます。dtrace は、バッファー領域の割り当てに失敗すると、バッファーサイズを減らそうとするか、終了します。どちらになるかは、bufresize プロパティーの設定によります。

-c *cmd*
: 指定されたコマンド *cmd* を実行し、その実行が完了したら終了します。コマンド行に -c オプションが複数指定されている場合、dtrace は、すべてのコマンドの終了後に終了しますが、それぞれの子プロセスが終了するたびにその終了ス

テータスを報告します。最初のコマンドのプロセス ID は、マクロ変数 `$target` により、コマンド行または `-s` オプションを使って指定された任意のプログラムに渡されます。マクロ変数の詳細については、『『Solaris Dynamic Tracing Guide』』を参照してください。

`-C`
D プログラムをコンパイルする前に、それらのプログラムに対して C プリプロセッサ cpp(1) を実行します。`-D`、`-U`、`-I`、`-H` の各オプションを使えば、C プリプロセッサにオプションを渡すことができます。`-X` オプションを使えば、C 標準への準拠レベルを選択できます。C プリプロセッサの呼び出し時に D コンパイラによって定義される一連のトークンについては、`-X` を参照してください。

`-D` *name* [=*value*]
`-C` オプションで有効化された cpp(1) を呼び出す際に、*name* を定義します。等号 (=) と追加の *value* を指定した場合、名前に対応する値が代入されます。このオプションは、`cpp` が呼び出されるたびに、それに `-D` オプションを渡します。

`-e`
すべての要求をコンパイルし、匿名トレース状態を消費し終わったあと (`-a` オプション)、プローブを有効化する前に終了します。このオプションを `-a` オプションと組み合わせれば、匿名トレースデータを出力してから終了させることができます。このオプションを D コンパイラのオプションと組み合わせることもできます。この組み合わせを使用した場合、プログラムがコンパイルされるかどうかの検証は行われますが、それらのプログラムが実際に実行されたり対応する計測機能が有効化されたりすることはありません。

`-f` [[*provider*:] *module*:] *function* [[*predicate*] *action*]]
トレースまたは一覧表示 (`-l` オプション) の対象となる関数名を指定します。対応する引数には、プローブ記述形式 *provider:module:function*、*module:function*、*function* のいずれかを含めることができます。指定しなかったプローブ記述フィールドは空になります。この場合、これらのフィールドの値とは関係なくすべてのプローブが選択されます。*function* 以外の修飾子を指定しなかった場合、対応する *function* を持つすべてのプローブが選択されます。`-f` の引数の後にはオプションで、D プローブ節を付加することもできます。`-f` オプションは、コマンド行で一度に複数指定できます。

`-F`
関数の開始 (entry) と終了 (return) を識別することにより、トレース出力をひとつにまとめます。関数の entry プローブの報告はインデントされ、 `->` の後ろに出力されます。関数終了プローブのレポートはインデント解除され、`<-` の後ろに出力されます。システムコール開始プローブのレポートはインデントされ、`=>` の後ろに出力されます。システムコール終了プローブのレポートはインデント解除され、`<=` の後ろに出力されます。

`-G`
埋め込まれた DTrace プログラムを含む ELF ファイルを生成します。プログラム内に指定された DTrace プローブは、ELF オブジェクトの内側に保存されます。このオブジェクトは再配置可能であり、別のプログラムにリンクできま

す。-o オプションが指定されている場合、ELF ファイルは、このオペランドの引数として指定されたパス名で保存されます。-o オプションが存在しておらず、かつ *filename*.d という名前のファイルに DTrace プログラムが格納されている場合、ELF ファイルは、*filename*.o という名前で保存されます。それ以外の場合、ELF ファイルは名前 d.out として保存されます。

-H
-C オプションで有効化された cpp(1) を呼び出す際に、インクルードされたファイルのパス名を出力します。このオプションは、cpp が呼び出されるたびに、それに -H オプションを渡します。その結果、一連のパス名が 1 行に 1 つずつ、標準エラー出力に出力されます。

-h
指定したプロバイダ定義内のプローブに対応するマクロを格納するヘッダーファイルを生成します。このオプションは、あとで -G オプションとともに使用するほかのソースファイルによってインクルードされるヘッダーファイルを生成する場合に使用してください。-o オプションが存在している場合、ヘッダーファイルはこのオプションの引数として指定されたパス名を使って保存されます。-o オプションが存在せず、かつ DTrace プログラムが *filename*.d というファイルに格納されている場合、ヘッダーファイルは名前 *filename*.h として保存されます。

-i *probe-id*[[*predicate*] *action*]
トレースまたは一覧表示 (-l オプション) の対象となるプローブ ID (*probe-id*) を指定します。プローブ ID を指定する際には、dtrace -l で表示される 10 進整数を使用します。-i の引数には、後ろに任意の D プローブ節を 1 つ付加できます。-i オプションは一度に複数指定できます。

-I *path*
-C オプションで有効化された cpp(1) を呼び出す際に、指定されたディレクトリ *path* を #include ファイルの検索パスに追加します。このオプションは、cpp の呼び出しのたびに -I オプションを渡します。指定された *path* は、検索パス内のデフォルトディレクトリリストの前に挿入されます。

-L *path*
指定されたディレクトリ *path* を、DTrace ライブラリの検索パスに追加します。DTrace ライブラリは、D プログラムを記述する際に使用可能な共通定義を格納する目的で使用されます。このオプションで指定された *path* は、デフォルトのライブラリ検索パスの後ろに追加されます。

-l
プローブを有効化しないで一覧表示します。-l オプションが指定された場合、dtrace は、-P、-m、-f、-n、-i、-s の各オプションを使って指定された記述に一致するプローブのレポートを生成します。これらのオプションが 1 つも指定されなかった場合、このオプションはすべてのプローブを一覧表示します。

-m [[*provider:*] *module:* [[*predicate*] *action*]]
トレースまたは一覧表示 (-l オプション) の対象となるモジュール名を指定します。対応する引数には、プローブ記述形式 *provider:module*、*module* のいずれかを含めることができます。指定しなかったプローブ記述フィールドは空になります。この場合、これらのフィールドの値とは関係なくすべてのプローブが選択されます。*module* 以外の修飾子を指定しなかった場合、対応する *module* を持つすべてのプローブが選択されます。-m の引数には、後ろに任意の D プローブ節を 1 つ付加できます。-m オプションは、コマンド行で一度に複数指定できます。

-n [[[*provider:*] *module:*] *function:*] *name* [[*predicate*] *action*]
トレースまたは一覧表示 (-l オプション) の対象となるプローブ名を指定します。対応する引数には、プローブ記述形式 *provider:module:function:name*、*module:function:name*、*function:name*、*name* のいずれかを含めることができます。指定しなかったプローブ記述フィールドは空になります。この場合、これらのフィールドの値とは関係なくすべてのプローブが選択されます。*name* 以外の修飾子を指定しなかった場合、対応する *name* を持つすべてのプローブが選択されます。-n の引数には、後ろに任意の D プローブ節を 1 つ付加できます。-n オプションは、コマンド行で一度に複数指定できます。

-o *output*
-A、-G、-h、および -l オプション、またはトレースされたデータ自体に対する *output* ファイルを指定します。-A オプションが指定されていて -o オプションが指定されていない場合、デフォルトの出力ファイルは /kernel/drv/dtrace.conf になります。-G オプションが存在していて -s オプションの引数が *filename*.d の形式であり、かつ -o が存在していない場合、デフォルトの出力ファイルは *filename*.o になります。それ以外の場合、デフォルトの出力ファイルは d.out になります。

-o オプションを使って dtrace を連続して呼び出すと、dtrace は上書きするのではなく、出力ファイルに追加します。

-p *pid*
指定されたプロセス ID *pid* を獲得し、そのシンボルテーブルをキャッシュし、その実行が完了したら終了します。コマンド行に -p オプションを 2 つ以上指定した場合、dtrace は、プロセスが終了するたびに終了ステータスを報告し、すべてのコマンドが終了した時点で終了します。最初のプロセス ID は、マクロ変数 $target により、コマンド行または -s オプションを使って指定された任意の D プログラムに渡されます。マクロ変数の詳細については、『『Solaris Dynamic Tracing Guide』』を参照してください。

-P *provider* [[*predicate*] *action*]
トレースまたは一覧表示 (-l オプション) の対象となるプロバイダ名を指定します。残りのプローブ記述フィールド (module、function、name) は空になります。したがって、これらのフィールドの値とは関係なく、すべてのプローブが選択されます。-P の引数には、後ろに任意の D プローブ節を 1 つ付加できます。-P オプションは、コマンド行で一度に複数指定できます。

-q
非出力モードを設定します。dtrace は、指定されたオプションや D プログラムの条件に一致するプローブ数などのメッセージを抑制し、列ヘッダーの CPU ID や プローブ ID を出力しなくなるほか、出力に改行を挿入しなくなります。trace() や printf() といった D プログラム文によってトレースおよび書式設定されたデータのみが、標準出力に表示されます。

-s
指定された D プログラムのソースファイルをコンパイルします。-e オプションが存在している場合、プログラムのコンパイルは行われますが、計測機能は有効化されません。-l オプションが存在している場合、プログラムのコンパイルと一致するプローブの一覧表示は行われますが、計測機能は有効化されません。-e、-l、-G、-A のいずれも存在しない場合には、D プログラムによって指定された計測機能が有効化され、トレースが開始されます。

-S
D コンパイラの中間コードを表示します。D コンパイラは、個々の D プログラムごとに生成された中間コードに関するレポートを生成し、標準エラー出力に出力します。

-U *name*
-C オプションで有効化された cpp(1) を呼び出す際に、指定された *name* を未定義にします。このオプションは、cpp が呼び出されるたびに、それに -U オプションを渡します。

-v
冗長モードを設定します。-v オプションを指定すると、dtrace はプログラム安定性レポートを生成します。このレポートには、指定された D プログラムについて最低限のインタフェースの安定性レベルと依存性レベルが示されます。DTrace の安定性レベルについては、『『Solaris Dynamic Tracing Guide』』に詳しく記載されています。

-V
dtrace がサポートする D プログラミングインタフェースのバージョンのうち、もっとも高いバージョンを報告します。このバージョン情報が標準出力に出力されたあと、dtrace コマンドは終了します。DTrace のバージョン管理機能の詳細については、『『Solaris Dynamic Tracing Guide』』を参照してください。

-w
-s、-P、-m、-f、-n、-i のいずれかのオプションで指定された D プログラム内で、破壊アクションを使用できるようにします。-w オプションが指定されなかった場合、dtrace は、破壊アクションを含む D プログラムのコンパイルや有効化を許可しません。

-x *arg* [=*val*]
DTrace の実行時オプションまたは D コンパイラオプションを有効化または変更します。オプションのリストは、『『Solaris Dynamic Tracing Guide』』に記載さ

れています。ブール型のオプションを有効にするときは、その名前を指定します。値を持つオプションを設定するには、オプションの名前と値を等号 (=) で分離します。

-X a | c | s | t

オプション -C オプションで有効化された cpp(1) を呼び出すとき、ISO C 標準にどの程度まで準拠させるかを指定します。-X オプションの引数は、__STDC__ マクロの値やその存在有無に影響を与えますが、具体的にどのような影響を与えるかは、この引数の文字の値によります。

-X オプションでサポートされる引数は、次のとおりです。

a　デフォルト。ISO C + K&R 互換性拡張。ISO C が必要とするセマンティック変更を含みます。-X が指定されなかった場合、これがデフォルトのモードになります。-Xa オプションが指定されている場合、cpp を呼び出したときの定義済みマクロ __STDC__ の値はゼロになります。

c　準拠。ISO C に完全準拠。K&R C 互換性拡張機能はありません。cpp 呼び出し時に -Xc オプションが指定された場合、定義済みマクロ __STDC__ の値は 1 になります。

s　K&R C のみ。cpp 呼び出し時に -Xs オプションが指定された場合、マクロ __STDC__ は未定義になります。

t　移行。ISO C + K&R C 互換性拡張。ISO C が必要とするセマンティック変更は含みません。cpp 呼び出し時に -Xt オプションが指定された場合、定義済みマクロ __STDC__ の値は 0 になります。

-X オプションは D コンパイラが C プリプロセッサを呼び出す方法にのみ影響を与えるため、-Xa オプションと -Xt オプションは、D から見れば等価です。それでもその両方が提供されているのは、C 構築環境の設定を再利用しやすくするためです。

次の追加 C プリプロセッサ定義は、-X モードの内容にかかわらず、すべてのモードで常に指定され、有効になります。

- __sun
- __unix
- __SVR4
- __sparc (SPARC システム上でのみ)
- __sparcv9 (SPARC システム上で 64 ビットプログラムをコンパイルする場合のみ)
- __i386 (x86 システム上で 32 ビットプログラムをコンパイルする場合のみ)
- __amd64 (x86 システム上で 64 ビットプログラムをコンパイルする場合のみ)
- __‘*uname -s*‘_‘*uname -r*‘ (例: __SunOS_5_10)

- `__SUNW_D=1`
- `__SUNW_D_VERSION=0x`*MMmmmuuu*

  ここで、*MM* は 16 進のメジャーリリース値、*mmm* は 16 進のマイナーリリース値、*uuu* は 16 進のマイクロリリース値です。DTrace のバージョン管理の詳細については、『『Solaris Dynamic Tracing Guide』』を参照してください。

`-Z`
一致するプローブが 1 つも見つからないようなプローブ記述を許可します。`-Z` オプションが指定されない場合、D プログラムファイル内で指定されたプローブ記述 (`-s` オプション) またはコマンド行で指定されたプローブ記述 (`-P`、`-m`、`-f`、`-n`、または `-i` オプション) の中で、既知のどのプローブにも一致しないような記述を含むプローブ記述が見つかると、`dtrace` はエラーを報告し、処理を終了します。

オペランド

`dtrace` コマンド行に 0 個以上の追加引数を指定し、一連のマクロ変数 (`$1` や `$2` など) を定義することができます。これらの追加引数は、`-s` オプションを使って指定された D プログラム内またはコマンド行から指定された D プログラム内で使用できます。マクロ変数の使用については、『『Solaris Dynamic Tracing Guide』』に詳しく記載されています。

終了ステータス

次の終了値が返されます。

0　正常終了。

D プログラム要求の場合、終了ステータス `0` は、プログラムのコンパイル、プローブの有効化、匿名ステータスの取得、のいずれかが成功したことを示します。指定されたトレース要求でエラーや欠落が発生した場合でも、`dtrace` は `0` を返します。

1　エラーが発生しました。

D プログラム要求の場合、終了ステータス `1` は、プログラムのコンパイルが失敗したか、指定された要求に応じられなかったことを示します。

2　コマンド行に無効なオプションまたは引数が指定されました。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/dtrace
インタフェースの安定性	下記を参照。

コマンド行の構文は「確実」です。人間が読める形式の出力は「不確実」です。

関連項目

cpp(1), isainfo(1), ssh(1), libdtrace(3LIB), driver.conf(4), attributes(5), dtrace(7D)

『Solaris Dynamic Tracing Guide』

使用法

-p フラグを使用している場合、ターゲットプロセスを検査し、結果を報告している間、dtrace はターゲットプロセスを停止します。停止中のプロセスは何も実行できません。つまり、たとえば X サーバーが、X サーバーの制御下にあるウィンドウで実行されている dtrace によって検査される場合、proc ツールは更新できないウィンドウに結果を表示しようとするため、ウィンドウシステム全体がデッドロックになる可能性があります。このような場合は、ssh(1) を使用して別のシステムからログインし、問題のある proc ツールを強制終了すると、デッドロックが解消されます。

名前

dumpadm – configure operating system crash dump

形式

```
/usr/sbin/dumpadm [-nuy] [-c content-type] [-d dump-device]
     [-m mink | minm | min%] [-s savecore-dir]
     [-r root-dir] [-z on | off]
```

機能説明

The `dumpadm` program is an administrative command that manages the configuration of the operating system crash dump facility. A crash dump is a disk copy of the physical memory of the computer at the time of a fatal system error. When a fatal operating system error occurs, a message describing the error is printed to the console. The operating system then generates a crash dump by writing the contents of physical memory to a predetermined dump device, which is typically a local disk partition. The dump device can be configured by way of `dumpadm`. Once the crash dump has been written to the dump device, the system will reboot.

Fatal operating system errors can be caused by bugs in the operating system, its associated device drivers and loadable modules, or by faulty hardware. Whatever the cause, the crash dump itself provides invaluable information to your support engineer to aid in diagnosing the problem. As such, it is vital that the crash dump be retrieved and given to your support provider. Following an operating system crash, the `savecore(1M)` utility is executed automatically during boot to retrieve the crash dump from the dump device and write it to your file system in compressed form, to a file name `vmdump.`*X*, where *X* is an integer identifying the dump. Afterwards, `savecore(1M)` can be invoked on the same or another system to expand the compressed crash dump to a pair of files named `unix.`*X* and `vmcore.`*X*. The directory in which the crash dump is saved on reboot can be configured using `dumpadm`.

For systems with a UFS root file system, the default dump device is configured to be an appropriate swap partition. Swap partitions are disk partitions reserved as virtual memory backing store for the operating system. Thus, no permanent information resides in swap to be overwritten by the dump. See `swap(1M)`. For systems with a ZFS root file system, dedicated ZFS volumes are used for swap and dump areas. For further information about setting up a dump area with ZFS, see the ZFS Administration Guide. To view the current dump configuration, use the `dumpadm` command with no arguments:

```
example# dumpadm

      Dump content: kernel pages
       Dump device: /dev/dsk/c0t0d0s1 (swap)
Savecore directory: /var/crash
  Savecore enabled: yes
   Save compressed: yes
```

When no options are specified, `dumpadm` displays the current crash dump configuration. The example above shows the set of default values: the dump content is set to kernel memory pages only, the dump device is a swap disk partition, the directory for `savecore` files is set to `/var/crash/`. `savecore` is set to run automatically on reboot and save the crash dump in a compressed format.

When one or more options are specified, dumpadm verifies that your changes are valid, and if so, reconfigures the crash dump parameters and displays the resulting configuration. You must be root to view or change dump parameters.

Upon system installation, dumpadm establishes a dump device of sufficient size, based on kernel memory and other internal information, to accommodate a dump file. If you subsequently attempt to create a dump device that is too small to store the dump file, dumpadm issues an error message and the operation fails.

オプション

The following options are supported:

-c *content-type*

Modify the dump configuration so that the crash dump consists of the specified dump content. The content should be one of the following:

kernel

Kernel memory pages only.

all

All memory pages.

curproc

Kernel memory pages, and the memory pages of the process whose thread was currently executing on the CPU on which the crash dump was initiated. If the thread executing on that CPU is a kernel thread not associated with any user process, only kernel pages will be dumped.

-d *dump-device*

Modify the dump configuration to use the specified dump device. The dump device may one of the following:

dump-device

A specific dump device specified as an absolute pathname, such as /dev/dsk/*cNtNdNsN* when the system is running a UFS root file system. Or, specify a ZFS volume, such as /dev/zvol/dsk/rpool/dump, when the system is running a ZFS root file system.

swap

If the special token swap is specified as the dump device, dumpadm examines the active swap entries and selects the most appropriate entry to configure as the dump device. See swap(1M). Refer to the NOTES below for details of the algorithm used to select an appropriate swap entry. When the system is first installed with a UFS root file system, dumpadm uses the value for swap to determine the initial dump device setting. A given ZFS volume cannot be configured for both the swap area and the dump device.

-m *min*k | *min*m | *min*%

Create a minfree file in the current savecore directory indicating that savecore should maintain at least the specified amount of free space in the file system where the savecore directory is located. The min argument can be one of the following:

k
: A positive integer suffixed with the unit k specifying kilobytes.

m
: A positive integer suffixed with the unit m specifying megabytes.

%
: A % symbol, indicating that the minfree value should be computed as the specified percentage of the total current size of the file system containing the savecore directory.

The savecore command will consult the minfree file, if present, prior to writing the dump files. If the size of these files would decrease the amount of free disk space below the minfree threshold, no dump files are written and an error message is logged. The administrator should immediately clean up the savecore directory to provide adequate free space, and re-execute the savecore command manually. The administrator can also specify an alternate directory on the savecore command-line.

-n
: Modify the dump configuration to not run savecore automatically on reboot. This is not the recommended system configuration; if the dump device is a swap partition, the dump data will be overwritten as the system begins to swap. If savecore is not executed shortly after boot, crash dump retrieval may not be possible.

-r *root-dir*
: Specify an alternate root directory relative to which dumpadm should create files. If no -r argument is specified, the default root directory / is used.

-s *savecore-dir*
: Modify the dump configuration to use the specified directory to save files written by savecore. The directory should be an absolute path and exist on the system. If upon reboot the directory does not exist, it will be created prior to the execution of savecore. See the NOTES section below for a discussion of security issues relating to access to the savecore directory. The default savecore directory is /var/crash/.

-u
: Forcibly update the kernel dump configuration based on the contents of /etc/dumpadm.conf. Normally this option is used only on reboot when starting svc:/system/dumpadm:default, when the dumpadm settings from the previous boot must be restored. Your dump configuration is saved in the configuration file for this purpose. If the configuration file is missing or contains invalid values for any dump properties, the default values are substituted. Following the update, the configuration file is resynchronized with the kernel dump configuration.

-y
: Modify the dump configuration to automatically run savecore on reboot. This is the default for this dump setting. See NOTES.

`-z on | off`
Modify the dump configuration to control the operation of `savecore` on reboot. The options are `on`, to enable saving core files in a compressed format, and `off`, to automatically uncompress the crash dump file. The default is `on`, because crash dump files can be very large and require less file system space if saved in a compressed format.

使用例

例 1 Reconfiguring The Dump Device To A Dedicated Dump Device:

The following command reconfigures the dump device to a dedicated dump device:

```
example# dumpadm –d /dev/dsk/c0t2d0s2

          Dump content: kernel pages
           Dump device: /dev/dsk/c0t2d0s2 (dedicated)
     Savecore directory: /var/crash
       Savecore enabled: yes
        Save compressed: yes
```

終了ステータス

The following exit values are returned:

`0`
Dump configuration is valid and the specified modifications, if any, were made successfully.

`1`
A fatal error occurred in either obtaining or modifying the dump configuration.

`2`
Invalid command line options were specified.

ファイル

`/dev/dump`
Dump device.

`/etc/dumpadm.conf`
Contains configuration parameters for `dumpadm`. Modifiable only through that command.

savecore-directory`/minfree`
Contains minimum amount of free space for *savecore-directory*. See `savecore(1M)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`svcs(1)`, `uname(1)`, `savecore(1M)`, `svcadm(1M)`, `swap(1M)`, `attributes(5)`, `smf(5)`

注意事項

The system crash dump service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/dumpadm:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

Dump Device Selection

When the special swap token is specified as the argument to dumpadm -d the utility will attempt to configure the most appropriate swap device as the dump device. dumpadm configures the largest swap block device as the dump device; if no block devices are available for swap, the largest swap entry is configured as the dump device. If no swap entries are present, or none can be configured as the dump device, a warning message will be displayed. While local and remote swap files can be configured as the dump device, this is not recommended.

Dump Device/Swap Device Interaction (UFS File Systems Only)

In the event that the dump device is also a swap device, and the swap device is deleted by the administrator using the swap -d command, the swap command will automatically invoke dumpadm -d swap in order to attempt to configure another appropriate swap device as the dump device. If no swap devices remain or none can be configured as the dump device, the crash dump will be disabled and a warning message will be displayed. Similarly, if the crash dump is disabled and the administrator adds a new swap device using the swap -a command, dumpadm -d swap will be invoked to re-enable the crash dump using the new swap device.

Once dumpadm -d swap has been issued, the new dump device is stored in the configuration file for subsequent reboots. If a larger or more appropriate swap device is added by the administrator, the dump device is not changed; the administrator must re-execute dumpadm -d swap to reselect the most appropriate device fom the new list of swap devices.

Minimum Free Space

If the dumpadm -m option is used to create a minfree file based on a percentage of the total size of the file system containing the savecore directory, this value is not automatically recomputed if the file system subsequently changes size. In this case, the administrator must re-execute dumpadm -m to recompute the minfree value. If no such file exists in the savecore directory, savecore will default to a free space threshold of one megabyte. If no free space threshold is desired, a minfree file containing size 0 can be created.

Security Issues

If, upon reboot, the specified savecore directory is not present, it will be created prior to the execution of savecore with permissions 0700 (read, write, execute by owner only) and owner root. It is recommended that alternate savecore directories also be created with similar permissions, as the operating system crash dump files themselves may contain secure information.

Default for savecore

System installation software might reserve a dedicated dump device (for example, a disk slice or a ZFS volume). In such a case, the dumpadm default can be set to -n, meaning that savecore does not run automatically when the system reboots. A crash image will be preserved on the dump device. Run /usr/sbin/savecore manually as root to retrieve the crash image and copy it to a file under /var/crash. The crash image will remain on the dump device until overwritten by a later one.

名前

editmap – query and edit single records in database maps for sendmail

形式

```
editmap -C file [-N] [-f] [-q | -u | -x] maptype mapname key
     ["value"]...
```

機能説明

The `editmap` command queries or edits one record in a database maps used by the keyed map lookups in `sendmail(1M)`. Arguments are passed on the command line and output (for queries) is directed to standard output.

Depending on how it is compiled, `editmap` handles up to three different database formats, selected using the *maptype* parameter. See `OPERANDS`.

If the `TrustedUser` option is set in the `sendmail` configuration file and `editmap` is invoked as root, the generated files are owned by the specified `TrustedUser`.

オプション

The following options are supported:

- `-C` *file* — Use the specified `sendmail` configuration file (*file*) to look up the `TrustedUser` option.
- `-f` — Disable the folding of all upper case letters in the key to lower case. Normally, all upper case letters in the key are folded to upper case. This is intended to mesh with the `-f` flag in the `K` line in `sendmail.cf`. The value is never case folded.
- `-N` — Include the null byte that terminates strings in the map (for alias maps).
- `-q` — Query the map for the specified key. If found, print value to standard output and exit with `0`. If not found then print an error message to `stdout` and exit with `EX_UNAVAILABLE`.
- `-u` — Update the record for key with value or inserts a new record if one doesn't exist. Exits with `0` on success or `EX_IOERR` on failure.
- `-x` — Delete the specific key from the map. Exit with `0` on success or `EX_IOERR` on failure.

オペランド

The following operands are supported:

key — The left hand side of a record.

Each record is of the form:

key value

key and *value* are separated by white space.

mapname — File name of the database map being created.

maptype — Specifies the database format. The following *maptype* parameters are available:

- `dbm` — Specifies DBM format maps.
- `btree` — Specifies B-Tree format maps.

`hash` Specifies hash format maps.

value The right hand side of a record.

Each record is of the form:

key value

key and *value* are separated by white space.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

関連項目

`makemap(1M)`, `sendmail(1M)`, `attributes(5)`

名前

edquota – edit user quotas for ufs file system

形式

```
edquota [-p proto_user] username...
```

```
edquota -t
```

機能説明

`edquota` is a quota editor. One or more users may be specified on the command line. For each user a temporary file is created with an ASCII representation of the current disk quotas for that user for each mounted ufs file system that has a `quotas` file, and an editor is then invoked on the file. The quotas may then be modified, new quotas added, etc. Upon leaving the editor, `edquota` reads the temporary file and modifies the binary quota files to reflect the changes made.

The editor invoked is `vi(1)` unless the `EDITOR` environment variable specifies otherwise.

Only the super-user may edit quotas. In order for quotas to be established on a file system, the root directory of the file system must contain a file, owned by root, called `quotas`. (See `quotaon(1M)`.)

proto_user and *username* can be numeric, corresponding to the UID of a user. Unassigned UIDs may be specified; unassigned names may not. In this way, default quotas can be established for users who are later assigned a UID.

If no options are specified, the temporary file created will have one or more lines of the format, where a block is considered to be a 1024 byte (1K) block:

```
fs mount_point blocks (soft =number, \
     hard =number ) inodes (soft =number, \
     hard =number)
```

The *number* fields may be modified to reflect desired values.

オプション

The following options are supported:

`-p` Duplicate the quotas of the *proto_user* specified for each *username* specified. This is the normal mechanism used to initialize quotas for groups of users.

`-t` Edit the soft time limits for each file system. If the time limits are zero, the default time limits in `/usr/include/sys/fs/ufs_quota.h` are used. The temporary file created will have one or more lines of the form

fs *mount_point* blocks time limit = *number tmunit*, files time limit = *number tmunit*

tmunit may be one of “month", “week", “day", “hour", “min" or “sec"; characters appended to these keywords are ignored, so you may write “months" or “minutes" if you prefer. The *number* and *tmunit* fields may be modified to set desired values. Time limits are printed in the greatest possible time unit such that the value is greater than or equal to one. If “default" is printed after the *tmunit*, this indicates that the value shown is zero (the default).

使用法

See largefile(5) for the description of the behavior of edquota when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

quotas	quota file at the file system root
/etc/mnttab	table of mounted file systems

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

vi(1), quota(1M), quotacheck(1M), quotaon(1M), repquota(1M), attributes(5), largefile(5), quotactl(7I)

注意事項

All UIDs can be assigned quotas.

名前

eeprom – EEPROM 表示およびロードユーティリティー

形式

`/usr/sbin/eeprom [-] [-f device] [parameter[=value]]`

機能説明

`eeprom` は、EEPROM 内のパラメータの値を表示または変更します。パラメータは、指定された順に処理されます。`eeprom` は、*parameter* と *value* の組み合わせを処理する際には、その指定された変更を EEPROM に対して行い、そうでない場合はその *parameter* の値を表示します。パラメータの指定子が 1 つも指定されなかった場合、`eeprom` はすべての EEPROM パラメータの値を表示します。「–」(ハイフン) フラグは、標準入力からパラメータと値を読み取ることを指定します (行ごとに *parameter* または *parameter=value* を 1 つずつ)。

EEPROM の内容を変更できるのは、スーパーユーザーだけです。

`eeprom` は、EEPROM のチェックサムを検証し、不正な場合にはメッセージを表示します。

platform-name はプラットフォームの実装名であり、この名前を確認するには、`uname(1)` の `-i` オプションを使用します。

SPARC

SPARC ベースのシステムでは、`eeprom` の `security-mode`、`security-password`、および `security-#badlogins` プロパティーを使用してファームウェアのパスワード保護が実装されています。

x86

EEPROM の記憶領域は、プラットフォーム固有のブート領域に存在するファイルを使用してシミュレートされます。`/boot/solaris/bootenv.rc` ファイルが EEPROM の記憶領域をシミュレートします。

x86 ベースのシステムでは通常、システムの BIOS 内でパスワード保護が実装されているため、`eeprom` プログラムにはパスワード保護のサポートはありません。x86 ベースのシステムで `security-mode`、`security-password`、および `security-#badlogins` プロパティーを設定することも可能ですが、それらのプロパティーは、x86 ベースのシステムでは特別な意味や動作は持ちません。

オプション

`-f` *device*
: *device* を EEPROM デバイスとして使用します。

オペランド

x86 のみ

acpi-user-options
: 電源管理仕様 Advanced Configuration and Power Interface (ACPI) の使用を制御する構成変数。この変数で使用可能な値は、使用する Solaris オペレーティングシステムのリリースによって異なります。

  Solaris 10 と Solaris 11 のすべてのリリースで、値 `0x0` は、システム上で ACPI が使用可能な場合にその使用が試みられることを意味します。値 `0x2` を使用すると、ACPI の使用が無効化されます。

Solaris 10 1/06 リリースの場合、値 0x8 は、システム上で ACPI が使用可能な場合に Solaris 10 の以前のリリースと互換性のあるモードで ACPI の使用が試みられることを意味します。Solaris 10 1/06 のデフォルトは 0x8 です。

Solaris 10 の 1/06 以降のリリースおよび Solaris 11 のデフォルトは、0x0 です。

ほとんどのユーザーは、安全にデフォルト値を受け入れることができます。デフォルト値では、ACPI が使用可能な場合には有効化されます。Solaris 1/06 以降の Solaris リリースで ACPI の使用に関わる問題が発生していると考えられる場合には、まず値 0x8 を試し、それで満足する結果が得られないときには、0x02 を使用することをお勧めします。

console
コンソールデバイスを指定します。指定できる値は、ttya、ttyb、text、graphics、および force-text です。text モードでは、コンソールの出力はフレームバッファーに送信され、入力はキーボードから取得されます。テキストモードのバリエーションである graphics は、キーが押されるか、あるいはコンソールログイン、sulogin、または kmdb によってコンソールの対話処理が必要になるまで、アニメーション付きのイメージを表示します。text の追加のバリエーションである force-text は、VGA アダプタを VGA テキストモードに設定して、ビットマップデバイスとして使用しないようにします。このプロパティーが存在しない場合、コンソールデバイスは input-device と output-device で指定されたデバイスになります。console プロパティー、input-device と output-device のプロパティーペアのどちらも存在しない場合、コンソールはデフォルトでフレームバッファーとキーボードになります。

screen-*#columns* screen-*#rows*
ビットマップデバイスで screen が graphics または text に設定されている場合、screen-*#columns* および screen-*#rows* によってテキストの列と行を必要な数だけ指定できます。それらはデフォルトでそれぞれ 80 と 24 に設定されます。

Nvram Configuration Parameters

OpenBoot システムの中には、一部のパラメータをサポートしていないものもあります。デフォルトは、システムや PROM のリビジョンによって異なります。システムのデフォルトを確認するには、ok (OpenBoot) プロンプトで printenv コマンドを入力し、その「Default Value」列の出力を参照してください。

auto-boot?
true の場合、電源投入後やリセット後に自動的にブートします。デフォルトは true です。x86 の場合、このパラメータは GRUB メニューファイルによって制御されます。installgrub(1M) を参照してください。

ansi-terminal?
端末エミュレータの動作を制御するために使用される構成変数。値 false が設定されると、端末エミュレータは ANSI エスケープシーケンスを解釈しなくなり、代わりにその情報を出力デバイスにエコーするようになります。デフォルトは true です。

boot-args
: ブートサブシステムに渡される引数の文字列を保持します。たとえば、カスタム JumpStart インストールを要求するには、`boot-args=' - install dhcp'` を使用します。`boot(1M)`、`kmdb(1)`、および `kernel(1M)` を参照してください。

boot-command
: `auto-boot?` が `true` の場合に実行されるコマンド。デフォルトは `boot` です。

boot-device
: ブート元となるデバイス。*boot-device* には、0 個以上のデバイス指定子をスペースで区切って含めることができます。各デバイス指定子は、PROM デバイス別名、PROM デバイスパスのいずれかになります。ブート PROM はリスト内の連続する各デバイス指定子のオープンを順次試みますが、その際、先頭のデバイス指定子から順に試みます。最初にオープンに成功したデバイス指定子が、ブート元のデバイスとして使用されます。デフォルトは `disk net` です。

boot-file
: ブートするファイル (空の文字列を指定した場合は二次ブートプログラムがデフォルトを選択する)。デフォルトは空の文字列です。

boot-from
: ブートデバイスとブートファイル (OpenBoot PROM バージョン 1.*x* のみ)。デフォルトは `vmunix` です。

boot-from-diag
: 診断用のブートデバイスとブートファイル (OpenBoot PROM バージョン 1.*x* のみ)。デフォルトは `le( )unix` です。

boot-ncpus
: システムブート時に使用されるプロセッサの数を制御する構成変数。デフォルトでは、システムブート時に、サポートされる最大数のプロセッサが使用されます。

com*X*-noprobe
: *X* はシリアルポートの番号。シリアルポート *X* でのデバイスの検索を行いません。

diag-device
: 診断ブート元デバイス。デフォルトは `net` です。

diag-file
: 診断モードのブート元となるファイル。デフォルトは空の文字列です。

diag-level
: 診断レベル。値には `off`、`min`、`max`、`menus` が含まれます。プラットフォーム固有の値が追加されている可能性があります。`off` に設定すると、POST は呼び出されません。POST が呼び出される場合、この値が POST の引数として渡され、そこで解釈されます。デフォルトは `platform-dependent` です。

diag-switch?
: true の場合、診断モードで実行されます。デフォルトは、大部分のデスクトップシステムでは false、大部分のサーバーでは true になります。

error-reset-recovery
: エラーリセットトラップ後に復旧します。デフォルトは、プラットフォーム固有の設定になります。

この変数をサポートするプラットフォームでは、watchdog-reboot?watchdog-sync?、redmode-reboot?、redmode-sync?、sir-sync?、xi の各パラメータがこの変数で置き換えられます。

オプションは、次のとおりです。

none
: リセットトラップを記述するメッセージを出力したあと、OpenBoot PROM のユーザーインタフェース、つまり OK プロンプトを表示します。

sync
: リセットトラップ後に OpenBoot PROM の sync ワードを呼び出します。これは、一部のプラットフォームでは、外部強制リセット (XIR) トラップ後の none として処理されることがあります。

boot
: リセットトラップ後にリブートします。これは、一部のプラットフォームでは、XIR トラップ後の none として処理されることがあります。

fcode-debug?
: true の場合、プラグインデバイス FCode の名前パラメータを含めます。デフォルトは false です。

hardware-revision
: システムのバージョン情報。

input-device
: 電源投入時に使用される入力デバイス (通常は keyboard、ttya、または ttyb のいずれか)。デフォルトは keyboard です。

keyboard-click?
: true の場合、キーボードクリックを有効にします。デフォルトは false です。

keyboard-layout
: 非自己識別キーボード (タイプ 7c) のレイアウト名を指定する文字列。使用可能なレイアウト名のリストを取得するには、kbd -s を呼び出します。kbd(1) を参照してください。

keymap
: カスタムキーボードのキーマップ。

last-hardware-update
: システムの更新情報。

load-base
: クライアントプログラムのデフォルトのロードアドレス。デフォルトの値は `16384` です。

local-mac-address?
: true の場合、ネットワークドライバは、システムの MAC アドレスではなく独自の MAC アドレスを使用します。デフォルトは `false` です。

mfg-mode
: POST 用の製造モード引数。指定可能な値は `off` または `chamber` です。この値は POST への引数として渡されます。デフォルトは `off` です。

mfg-switch?
: true の場合、STOP-A による割り込みが発生するまで、システムの自己診断テストを繰り返し行います。デフォルトは `false` です。

nvramrc
: NVRAMRC の内容。デフォルトは空です。

network-boot-arguments
: ネットワークブート時に PROM によって使用される引数。デフォルトは空の文字列です。`network-boot-arguments` を使用すると、使用されるブートプロトコル (RARP/DHCP) や、プロセス中に使用されるさまざまなシステム情報を指定できます。

  ネットワークブートでサポートされる引数の構文は、次のとおりです。

  `[`*protocol*`,] [`*key*`=`*value*`,]*`

  すべての引数は省略可能であり、指定する順序にも制限はありません。ただし、リストの末尾でない限り、引数の後にはコンマが必要です。ここで指定した引数は、すべてのデフォルト値よりも優先されるほか、DHCP を使ってブートする場合には、それらの引数に対応する DHCP サーバーから提供される構成情報よりも優先されます。

  前述の *protocol* には、使用するアドレス検索プロトコルを指定します。

  key=*value* 属性ペアには、以下の構成パラメータを指定します。

  `tftp-server`
  : TFTP サーバーの IP アドレス。

  `file`
  : TFTP を使ってダウンロードするファイルの名前、または WANブートの URL。

  `host-ip`
  : クライアントの IP アドレス (ドット区切り 10 進表記)。

  `router-ip`
  : デフォルトのルーターの IP アドレス (ドット区切り 10 進表記)。

`subnet-mask`
サブネットマスク (ドット区切り 10 進表記)。

`client-id`
DHCP クライアント識別子。

`hostname`
DHCP トランザクションで使用するホスト名。

`http-proxy`
HTTP プロキシサーバー指定 (IPADDR[: PORT])。

`tftp-retries`
TFTP の最大リトライ回数。

`dhcp-retries`
DHCP の最大リトライ回数。

パラメータが 1 つも指定されなかった場合 (つまり、`network-boot-arguments` が空の文字列だった場合)、PROM はプラットフォーム固有のデフォルトアドレス検索プロトコルを使用します。

プロトコルパラメータを指定せずにほかの構成パラメータを指定した場合、暗黙的に手動構成が選択されます。

手動構成の場合、ブートに必要なすべての情報をクライアントに提供する必要があります。手動構成を使用する場合、PROM が二次ブートプログラムをロードするために必要な情報については、`network-boot-arguments` に指定する必要がありますが、二次ブートプログラムで必要になる情報については、`boot` プログラムの引数として指定することも、`boot` プログラムの対話型コマンドインタプリタ経由で指定することもできます。

手動構成の使用時に PROM で必要になる情報は、ブート対象クライアントの IP アドレス、ブートファイルの名前、およびブートファイルイメージを提供するサーバーのアドレスです。ネットワーク構成によっては、使用するサブネットマスクやデフォルトルーターのアドレスも指定しなければならない場合があります。

oem-banner
カスタム OEM バナー (`oem-banner?` を `true` に設定すると有効になる)。デフォルトは空の文字列です。

oem-banner?
`true` の場合、カスタム OEM バナーを使用します。デフォルトは `false` です。

oem-logo
バイト配列のカスタム OEM ロゴ (`oem-logo?` を `true` に設定すると有効になる)。16 進数で表示されます。

oem-logo?
: `true` の場合、カスタム OEM ロゴを使用します (それ以外の場合は Sun のロゴを使用する)。デフォルトは `false` です。

pci-mem64?
: true の場合、OpenBoot PROM は、64 ビットアドレスをサポート可能な PCI デバイスに 64 ビット PCI メモリーアドレスを割り当てます。

この変数は SPARC プラットフォームでのみ使用可能であり、省略可能です。SunOS の一部のバージョンでは PCI `MEM64` アドレスがサポートされていないため、それらのバージョンで OpenBoot PROM が PCI `MEM64` アドレスを割り当てると、予期しない方法で処理が失敗します。

デフォルト値はシステムに依存します。この変数が存在する場合、そのデフォルト値は、特定のプラットフォームに付属する SunOS のバージョンのうちでもっとも低いものに適した値になります。

output-device
: 電源投入時に使用される出力デバイス (通常は `screen`、`ttya`、または `ttyb` のいずれか)。デフォルトは `screen` です。

redmode-reboot?
: redmode リセットトラップ後にリブートする場合は、`true` を指定します。デフォルトは `true` です。(Sun Enterprise 10000 のみ。)

redmode-sync?
: redmode リセットトラップ後に OpenBoot PROM の `sync` ワードを呼び出す場合は、`true` を指定します。デフォルトは `false` です。(Sun Enterprise 10000 のみ。)

rootpath
: オペレーティングシステムのルートデバイスを指定します。

sbus-probe-list
: どの SBus スロットをどのような順序で検索するかを指定します。デフォルトは `0123` です。

screen-#columns
: 画面上の列の数 (1 行の文字数)。デフォルトは `80` です。

screen-#rows
: 画面上の行の数 (行数)。デフォルトは `34` です。

scsi-initiator-id
: ホストアダプタの SCSI バスアドレスで、範囲は 0 から 7 です。デフォルトは `7` です。

sd-targets
: SCSI ディスクユニットをマップします (OpenBoot PROM バージョン 1.*x* のみ)。デフォルトは `31204567` ですが、これは、ユニット 0 がターゲット 3 にマップされ、ユニット 1 がターゲット 1 にマップされ、といったようになることを意味します。

security-#badlogins
: セキュリティーパスワードの不正な試行回数。x86 ベースのシステムでは、このプロパティーは特別な意味や動作を持ちません。

security-mode
: ファームウェアのセキュリティーレベル (オプションは、`none`、`command`、または `full`)。`command` または `full` に設定すると、システムから PROM セキュリティーパスワードの入力を求められます。デフォルトは `none` です。x86 ベースのシステムでは、このプロパティーは特別な意味や動作を持ちません。

security-password
: ファームウェアのセキュリティーパスワード (表示されない)。`security-mode` が `command` または `full` に設定された場合にのみ設定可能です。x86 ベースのシステムでは、このプロパティーは特別な意味や動作を持ちません。

```
example# eeprom security-password=
Changing PROM password:
New password:
Retype new password:
```

selftest-#megs
: テストする RAM のメガバイト数。`diag-switch?` を `true` に設定した場合は無視されます。デフォルトは `1` です。

sir-sync?
: ソフトウェア起動リセット (SIR) トラップ後に OpenBoot PROM の `sync` ワードを呼び出す場合は、`true` を指定します。デフォルトは `false` です。(Sun Enterprise 10000 のみ。)

skip-vme-loopback?
: `true` の場合、POST は VMEbus ループバックテストを行いません。デフォルトは `false` です。

st-targets
: SCSI テープユニットをマップします (OpenBoot PROM バージョン 1.*x* のみ)。デフォルトは `45670123` ですが、これは、ユニット 0 がターゲット 4 にマップされ、ユニット 1 がターゲット 5 にマップされ、といったようになることを意味します。

sunmon-compat?
: `true` の場合、制限モニタープロンプト (>) を表示します。デフォルトは `false` です。

testarea
: 読み取り/書き込みテストで使用可能な 1 バイトのスクラッチフィールド。デフォルトは `0` です。

tpe-link-test?
: 内蔵されている、より対線 Ethernet に対する 10baseT リンクテストを有効にします。デフォルトは `true` です。

ttya-mode
: TTYA (ボーレート、ビット数、パリティー、ストップ数、ハンドシェーク)。デフォルトは `9600,8,n,1,–` です。

各フィールドは次のとおりです (左から右へ)。

ボーレート:
: 110、300、1200、4800、9600 . . .

データビット数:
: 5、6、7、8

パリティー:
: n(none)、e(even)、o(odd)、m(mark)、s(space)

ストップビット数:
: 1、1.5、2

ハンドシェーク:
: –(none)、h(hardware:rts/cts)、s(software:xon/xoff)

ttyb-mode
: TTYB (ボーレート、ビット数、パリティー、ストップ数、ハンドシェーク)。デフォルトは `9600,8,n,1,–` です。

各フィールドは次のとおりです (左から右へ)。

ボーレート:
: 110、300、1200、4800、9600 . . .

データビット数:
: 5、6、7、8

ストップビット数:
: 1、1.5、2

パリティー:
: n(none)、e(even)、o(odd)、m(mark)、s(space)

ハンドシェーク:
: –(none)、h(hardware:rts/cts)、s(software:xon/xoff)

ttya-ignore-cd
: true の場合、オペレーティングシステムは TTYA でのキャリア検出を無視します。デフォルトは true です。

ttyb-ignore-cd
: true の場合、オペレーティングシステムは TTYB でのキャリア検出を無視します。デフォルトは true です。

ttya-rts-dtr-off
: true の場合、オペレーティングシステムは TTYA で DTR と RTS を表明しません。デフォルトは false です。

ttyb-rts-dtr-off
: true の場合、オペレーティングシステムは TTYB で DTR と RTS を表明しません。デフォルトは false です。

use-nvramrc?
: true の場合、システムの起動中に NVRAMRC 内のコマンドを実行します。デフォルトは false です。

verbosity
: PROM メッセージの詳細レベルを制御します。debug、max、normal、min、または none のいずれかを指定できます。デフォルトは normal です。

version2?
: true の場合、ハイブリッド (1.*x*/2.*x*) PROM がバージョン 2.*x* で起動されます。デフォルトは true です。

watchdog-reboot?
: true の場合、ウォッチドッグリセット後にリブートします。デフォルトは false です。

watchdog-sync?
: ウォッチドッグリセットトラップ後に OpenBoot PROM の sync ワードを呼び出す場合は、true を指定します。デフォルトは false です。(Sun Enterprise 10000 のみ。)

xir-sync?
: XIR トラップ後に OpenBoot PROM の sync ワードを呼び出す場合は、true を指定します。デフォルトは false です。(Sun Enterprise 10000 のみ。)

使用例

例 1　RAM のメガバイト数の変更

次の例は、システムがテストする RAM のメガバイト数を 1 から 2 に変更する方法を示しています。

```
example# eeprom selftest-#megs
selftest-#megs=1

example# eeprom selftest-#megs=2
```

例 1　RAM のメガバイト数の変更　　*(続き)*

```
example# eeprom selftest-#megs
selftest-#megs=2
```

例 2　auto-boot? パラメータへの true の設定

次の例は、auto-boot? パラメータを true に設定する方法を示しています。

```
example# eeprom auto-boot?=true
```

eeprom コマンドをユーザーモードで実行する場合、末尾に疑問符 (?) の付くパラメータを二重引用符 (“ “) で囲み、その疑問符がシェルによって解釈されないようにする必要があります。疑問符がシェルによって解釈されないようにするには、疑問符の前にエスケープ文字 (\) を付けてもかまいません。

```
example% eeprom "auto-boot?"=true
```

例 3　network-boot-arguments の使用

DHCP をブートプロトコルとして使用し、ネットワークブート時にホスト名 abcd.example.com を使用するには、network-boot-arguments 内でこれらの値を次のように設定します。

```
example# eeprom network-boot-arguments="dhcp,hostname=abcd.example.com"
```

そして、次のコマンドを使用してブートします。

```
ok boot net
```

PROM コマンド行からネットワークブート引数を指定すると、network-boot-arguments の内容が無視されます。たとえば、network-boot-arguments が前述のように設定された状態で次の boot コマンドを実行します。

```
ok boot net:dhcp
```

このとき、DHCP は使用されますが、network-boot-arguments に指定された hostname はネットワークブート時に使用されません。

例 4　補助デバイスへのシステムコンソールの設定

次のコマンドは、デバイス /dev/term/a をシステムのコンソールデバイスとして割り当てます。そのような割り当ては、tip(1) を使用してホストへの tip 接続を確立する前に行います。

SPARC マシンの場合:

```
# eeprom output-device=/dev/term/a
```

x86 マシンの場合:

例 4　補助デバイスへのシステムコンソールの設定　　(続き)

```
# eeprom console=ttya
```

SPARC マシンの場合、コンソールを補助デバイスに割り当てるには、前述のコマンドで十分です。x86 マシンの場合はさらに、シリアル回線の特性を設定しなければいけない可能性がありますが、その方法については、そのマシンの BIOS のドキュメントを参照する必要があります。また、一部の x86 マシンでは、前述のデバイス a 以外のデバイスを使用する可能性もあります。たとえば、2 つ目のシリアル回線が存在する場合は、コンソールを ttyb に設定することもできます。

例 5　SPARC システムでの kmdb を使用したブートの指定

SPARC マシンが kmdb(1) を使用してブートするように指定するには、次のコマンドを入力します。

```
# eeprom boot-command="boot -k"
```

ファイル

/boot/solaris/bootenv.rc
　x86 マシンで eeprom 値を格納するファイル。

/dev/openprom
　デバイスファイル

/usr/platform/*platform-name*/sbin/eeprom
　プラットフォーム固有のバージョンの eeprom。*platform-name* を取得するには uname -i を使用します。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

kmdb(1), passwd(1), sh(1), svcs(1), tip(1), uname(1), boot(1M), kernel(1M), init(1M), svcadm(1M), attributes(5), smf(5)

『OpenBoot 3.x Command Reference Manual』

『ONC+ Developer's Guide』

名前

efdaemon – embedded FCode interpreter daemon

形式

```
/usr/lib/efcode/sparcv9/efdaemon [-d]
```

機能説明

efdaemon, the embedded FCode interpreter daemon, invokes the embedded FCode interpreter when the daemon receives an interpretation request. A new session of the interpreter is started for each unique request by invoking the script /usr/lib/efcode/efcode.

efdaemon is used on selected platforms as part of the processing of some dynamic reconfiguration events.

オプション

The following option is supported:

-d　Set debug output. Log debug messages as LOG_DEBUG level messages by using syslog(). See syslog(3C).

ファイル

/dev/fcode	FCode interpreter pseudo device, which is a portal for receipt of FCode interpretation requests
/usr/lib/efcode/efcode	Shell script that invokes the embedded FCode interpreter
/usr/lib/efcode/interpreter	Embedded FCode interpreter
/usr/lib/efcode/sparcv9/interpreter	Embedded FCode interpreter

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWefcx, SUNWefcux, system/embedded-fcode-interpreter, SUNWefclx

関連項目

svcs(1), prtconf(1M), svcadm(1M), syslog(3C), attributes(5), smf(5)

注意事項

The efdaemon service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/platform/sun4u/efdaemon:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

embedded_su – allow an application to prompt for credentials and execute commands as the super user or another user

形式

`/usr/lib/embedded_su [-] [`*`username`* `[arg...]]`

機能説明

The `embedded_su` command allows an application to prompt the user for security credentials and then use those credentials to execute a program as another user or role (see `rbac(5)` for information on role-based access control). The default *username* is `root` (super user).

`embedded_su` is identical to `su(1M)`, except that the user interaction is packaged in a form suitable for another program to interpret and display. Typically, `embedded_su` would be used to allow a graphical program to prompt for the super user password and execute a command as the super user, without requiring that the requesting program be run as the super user.

PROTOCOL

`embedded_su` implements a simple protocol over standard input, standard output, and standard error. This protocol consists of three phases, roughly corresponding to PAM initialization, the PAM dialog, and PAM completion.

Phase 1: Initialization

After starting `embedded_su`, the application must send an initialization block on `embedded_su`'s standard input. This block is a text block, as described under “Text Blocks”. There are currently no initialization parameters defined; the application should send an empty block by sending a line consisting solely of a period (.).

Phase 2: Conversation

`embedded_su` then emits zero or more conversation blocks on its standard output. Each conversation block may require zero or more responses.

A conversation block starts with a line consisting of the word `CONV`, followed by whitespace, followed by the number of messages in the conversation block as a decimal integer. The number of messages may be followed by whitespace and additional data. This data, if present, must be ignored.

Each message consists of a line containing a header followed by a text block, as described under “Text Blocks”. A single newline is appended to each message, allowing the message to end with a line that does not end with a newline.

A message header line consists of a PAM message style name, as described in `pam_start(3PAM)`. The message header values are:

`PAM_PROMPT_ECHO_OFF`	The application is to prompt the user for a value, with echoing disabled.
`PAM_PROMPT_ECHO_ON`	The application is to prompt the user for a value, with echoing enabled.
`PAM_ERROR_MSG`	The application is to display the message in a form appropriate for displaying an error.

PAM_TEXT_INFO — The application is to display the message in a form appropriate for general information.

The PAM message style may be followed by whitespace and additional data. This data, if present, must be ignored.

After writing all of the messages in the conversation block, if any of them were PAM_PROMPT_ECHO_OFF or PAM_PROMPT_ECHO_ON, embedded_su waits for the response values. It expects the response values one per line, in the order the messages were given.

Phase 3: Completion

After zero or more conversation blocks, embedded_su emits a result block instead of a conversation block.

Upon success, embedded_su emits a single line containing the word "SUCCESS". The word SUCCESS may be followed by whitespace and additional data. This data, if present, must be ignored.

Upon failure, embedded_su emits a single line containing the word "ERROR", followed by a text block as described under "Text Bocks". The text block gives an error message. The word ERROR may be followed by whitespace and additional data. This data, if present, must be ignored.

Text Blocks

Initialization blocks, message blocks, and error blocks are all text blocks. These are blocks of text that are terminated by a line containing a single period (.). Lines in the block that begin with a "." have an extra "." prepended to them.

Internationalization

All messages are localized to the current locale; no further localization is required.

SECURITY

embedded_su uses pam(3PAM) for authentication, account management, and session management. Its primary function is to export the PAM conversation mechanism to an unprivileged program. Like su(1M), the PAM configuration policy can be used to control embedded_su. The PAM service name used is "embedded_su".

embedded_su is almost exactly equivalent to su(1M) for security purposes. The only exception is that it is slightly easier to use embedded_su in writing a malicious program that might trick a user into providing secret data. For those sites needing maximum security, potentially at the expense of application functionality, the EXAMPLES section shows how to disable embedded_su.

使用例

In the following examples, left angle brackets (<<<) indicate a line written by embedded_su and read by the invoking application. Right angle brackets (>>>) indicate a line written by the application and read by embedded_su.

例 1 Executing a command with the Correct Password

The following example shows an attempt to execute “somecommand” as “someuser”, with the correct password supplied:

```
/usr/lib/embedded_su someuser -c somecommand
   >>>.
   <<<CONV 1
   <<<PAM_PROMPT_ECHO_OFF
   <<<Password:
   <<<.
   >>>[ correct password ]
   <<<SUCCESS
   [ somecommand executes  ]
```

例 2 Executing a command with the Incorrect Password

The following example shows an attempt to execute “somecommand” as “someuser”, with the incorrect password supplied:

```
/usr/lib/embedded_su someuser -c somecommand
   >>>.
   <<<CONV 1
   <<<PAM_PROMPT_ECHO_OFF
   <<<Password:
   <<<.
   >>>[ incorrect password ]
   [ delay ]
   <<<ERROR
   <<<embedded_su:Sorry
   <<<.
   [ exit ]
```

例 3 Message Examples

A `pam_message` structure with *msg_style* equal to `PAM_TEXT_INFO` and `msg` equal to “foo” produces:

```
PAM_TEXT_INFO
foo
.
```

A `pam_message` structure with *msg_style* equal to `PAM_ERROR_MESSAGE` and `msg` equal to “bar\n” produces:

```
PAM_ERROR_MESSAGE
bar
[ blank line ]
.
```

例 3 Message Examples (続き)

A pam_message structure with *msg_style* equal to PAM_ERROR_MESSAGE and msg equal to "aaa\nbbb" produces:

```
PAM_ERROR_MESSAGE
aaa
bbb
.
```

A pam_message structure with *msg_style* equal to PAM_TEXT_INFO and msg equal to "" produces:

```
PAM_TEXT_INFO
[ blank line ]
.
```

A pam_message structure with *msg_style* equal to PAM_TEXT_INFO and msg equal to NULL produces:

```
PAM_TEXT_INFO
.
```

例 4 Disabling embedded_su

To disable embedded_su, add a line to the /etc/pam.conf file similar to:

```
embedded_su  auth  requisite  pam_deny.so.1
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

su(1M), pam(3PAM), pam_start(3PAM), attributes(5), rbac(5)

名前

etrn – start mail queue run

形式

`etrn [-b] [-v] server-host [client-hosts]`

機能説明

SMTP's `ETRN` command allows an SMTP client and server to interact, giving the server an opportunity to start the processing of its queues for messages to go to a given host. This is meant to be used in start-up conditions, as well as for mail nodes that have transient connections to their service providers.

The `etrn` utility initiates an SMTP session with the host *server-host* and sends one or more `ETRN` commands as follows: If no *client-hosts* are specified, `etrn` looks up every host name for which `sendmail(1M)` accepts email and, for each name, sends an `ETRN` command with that name as the argument. If any *client-hosts* are specified, `etrn` uses each of these as arguments for successive `ETRN` commands.

オプション

The following options are supported:

`-b` System boot special case. Make sure localhost is accepting SMTP connections before initiating the SMTP session with server-host.

This option is useful because it prevents race conditions between `sendmail(1M)` accepting connections and *server-host* attempting to deliver queued mail. This check is performed automatically if no *client-hosts* are specified.

`-v` The normal mode of operation for `etrn` is to do all of its work silently. The `-v` option makes it verbose, which causes `etrn` to display its conversations with the remote SMTP server.

環境変数

No environment variables are used. However, at system start-up, `svc:/network/smtp:sendmail` reads `/etc/default/sendmail`. In this file, if the variable `ETRN_HOSTS` is set, `svc:/network/smtp:sendmail` parses this variable and invokes `etrn` appropriately. `ETRN_HOSTS` should be of the form:

```
"s1:c1.1,c1.2        s2:c2.1 s3:c3.1,c3.2,c3.3"
```

That is, white-space separated groups of *server:client* where *client* can be one or more comma-separated names. The *:client* part is optional. *server* is the name of the server to prod; a mail queue run is requested for each *client* name. This is comparable to running:

`/usr/lib/sendmail -qR client`

on the host *server*.

使用例

例 1 Using etrn

Inserting the line:

```
ETRN_HOSTS="s1.domain.com:clnt.domain.com s2.domain.com:clnt.domain.com"
```

例 1 Using etrn (続き)

in /etc/default/sendmail results in svc:/network/smtp:sendmail invoking etrn such that ETRN commands are sent to both s1.domain.com and s2.domain.com, with both having clnt.domain.com as the ETRN argument.

The line:

```
ETRN_HOSTS="server.domain.com:client1.domain.com,client2.domain.com"
```

results in two ETRN commands being sent to server.domain.com, one with the argument client1.domain.com, the other with the argument client2.domain.com.

The line:

```
ETRN_HOSTS="server1.domain.com server2.domain.com"
```

results in set of a ETRN commands being sent to both server1.domain.com and server2.domain.com; each set contains one ETRN command for each host name for which sendmail(1M) accepts email, with that host name as the argument.

ファイル

/etc/mail/sendmail.cf sendmail configuration file

/etc/default/sendmail Variables used by svc:/network/smtp:sendmail

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail
Interface Stability	Committed

関連項目

sendmail(1M), attributes(5)

RFC 1985

注意事項

Not all SMTP servers support ETRN.

名前

fbconfig – frame buffer configuration utility

形式

```
fbconfig [-dev device_file] [-help]

fbconfig -list

fbconfig -gui

fbconfig -res \?

fbconfig [-dev device_file] [-file machine | system]
     [device_specific_options] [-res video-mode] [defaults]
     [-prconf] [-prdid[raw] [parsed]] [-propt]
```

機能説明

`fbconfig` is the generic command line interface to query and/or to configure a frame buffer device. Depending on the command line options, `fbconfig` can invoke a GUI program, or it can invoke a device-specific configuration program. The choice of device configuration program depends both on the device (specified with `-dev`) and on the currently configured X server.

The options recognized by `fbconfig` are shown in the first three command forms in the `SYNOPSIS` section. The remaining forms illustrate options that may be passed to a device-specific configuration program that performs the actual operations. The interpretation of these options will depend upon the specific configuration program that is invoked. The options shown are supported by most device configuration programs.

If the `-dev` option is omitted, the default frame buffer (`/dev/fb` or `/dev/fb0`) is assumed.

オプション

The following options are supported by `fbconfig`:

`-dev` *device_file*

Specify the frame buffer device by its full pathname or simple filename. Pathnames of installed devices can be displayed using the `-list` option. If the `-dev` option is omitted, the default device is `/dev/fb`.

`-gui`

Invoke the Graphical User Interface (GUI). The GUI is available if the `SUNWdcm` package is installed. All other options are ignored.

The GUI can configure devices (as an alternative to the `fbconfig` command line) and can update the `Xservers` file without directly editing the file. The GUI allows the user that is logged in on the graphics device or devices to configure which graphics displays the window system should use, their screen layout (where they appear on the user's desktop), and screen properties (X attributes).

In addition, the GUI allows advanced users to create a new video format (resolution) that some graphics devices can select from `fbconfig` command line or from the device-dependent portion of the GUI. The GUI's online help explains all options and features.

`-help`
: Displays the supported `fbconfig` command line options, along with a brief explanation of each. Also displays the `-help` text of the device-specific configuration program, if any. The frame buffer device can be specified using the `-dev` option, otherwise the default is used. Other `fbconfig` options are ignored. This is the default `fbconfig` option.

`-list`
: Displays the pathnames of the installed frame buffer devices, the device model of each, and the configuration program that would be invoked for each device with the currently configured X server. Other `fbconfig` options are ignored.

```
Device File Name            Device Model     Config Program
----------------            ------------     --------------
/dev/fbs/kfb0               XVR-2500         SUNWkfb_config
/dev/fbs/kfb1               XVR-2500         SUNWkfb_config
/dev/fbs/nfb0 [a|b]         XVR-300          SUNWnfb_config
/dev/fbs/pfb0 [a|b]         XVR-100          SUNWpfb_config
```

The following options are commonly supported by the device-specific configuration programs:

`-defaults`
: Sets configuration options to their default values.

`-file machine | system`
: Specifies which `xorg.conf` configuration file to use.

`-prconf`
: Display the current configuration for the frame buffer and display device(s).

`-predid [raw] [parsed]`
: Display the E-EDID (Enhanced Extended Display Identification Data) information obtained from the display device(s), which must be online, connected to the frame buffer. The output will be raw hexadecimal and/or human-readable text. The default is `parsed`.

`-propt`
: Display the current software configuration.

`-res \?`
: Display the video modes (resolutions) that can be used with the `-res` *video-mode* option.

Other supported options are determined by the device-specific configuration program:

device-specific-options
: The syntax and descriptions of additional device-specific options are displayed in the `-help` output of `fbconfig` or the device-specific program. They are also contained in the man page for the device-specific program.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/graphics/fbconfig

関連項目

fbconf_xorg(1M), attributes(5)

Xserver(1), Xorg(1), Xsun(1), which are part of the OpenWindow, not the SunOS, man page collection.

Limitations

Because of limitations in the m64 kernel driver and related software, fbconfig (with the -prconf option) is unable to distinguish between a current depth of 24 or 8+24. The -propt option returns the depth specified in the OWconfig file, which will be in effect following the next restart of the window system. The xwininfo utility, usually shipped in the package containing frame buffer software (such as SUNWxwplt), reports current depth of a specified window.

名前

fbconf_xorg – configure frame buffer devices for Xorg

形式

```
/usr/lib/fbconfig/fbconf_xorg [-dev device-file]
     [-file machine | system | config-path]
     [-res video-mode [nocheck | noconfirm]]
     [device-specific-options]
     [-defaults] [-prconf] [-predid [raw] [parsed]] [-propt]
```

```
/usr/lib/fbconfig/fbconf_xorg [-dev device-file] [-prconf] [-propt]
```

```
/usr/lib/fbconfig/fbconf_xorg [-dev device-file] [-help] [-res ?]
```

機能説明

The `fbconf_xorg` utility configures Frame Buffer devices and some of the X11 window system defaults for Xorg by updating the `xorg.conf` configuration file. Users should not normally need to run `fbconf_xorg` directly, as it is executed by `fbconfig(1M)` when the Xorg server is in use.

The first `fbconf_xorg` command form shown in the `SYNOPSIS` section stores the specified option values in the `xorg.conf` configuration file. These settings are used to initialize the frame buffer device the next time the window system is run on that device. The persistence of these settings across window system sessions and system reboots is determined by the `xorg.conf` file.

The second and third command forms display information, and do not alter the `xorg.conf` file. The presence of the `-help` and/or `-res ?` options causes any other option (not shown in the third form) to be ignored.

You can configure only one frame buffer device at a time. To configure multiple devices, invoke the `fbconf_xorg` utility separately for each device.

Only frame buffer device options can be specified with `fbconf_xorg`. Use the normal window system options to specify default depth (see `svccfg(1M)`), default visual class, and so forth. Specify these as device modifiers on the command line as specified in Xserver(1).

You can specify which `xorg.conf` file to open. By default, `fbconf_xorg` opens the machine-specific file, `/etc/X11/xorg.conf`. Use the `-file` option to specify an alternate file. For example, the system-global file, `/usr/lib/X11/xorg.conf`, can be opened instead.

These standard `xorg.conf` files can be written only by the superuser or someone with the Desktop Configuration role.

オプション

The following options are supported for all frame buffer devices:

`-defaults`
: Sets configuration options for the specified device to their default values. This does not affect the `-res` video mode setting. See the device-specific portions of the `DEFAULTS` section below.

`-dev` *device-file*
Specifies the frame buffer device by either its full pathname or simple filename (for example, `/dev/fbs/efb0` or `efb0`). Pathnames of installed devices can be displayed using the `-list` option to `fbconfig(1M)`. If the `-dev` option is omitted, the default device, `/dev/fb`, is used.

`-file` `machine` | `system` | *config-path*
Specifies which `xorg.conf` file to open. If `machine` is specified, the machine-specific `/etc/X11/xorg.conf` file is opened. If system is specified, the global `/usr/lib/X11/xorg.conf` file is opened. The absolute pathname of a configuration file can be used instead. If the specified file does not exist and is to be updated, it is created. The file system that contains the `xorg.conf` file must be writeable by someone with superuser-like privileges. This option has no effect unless other options are specified. The default is `machine`.

`-help`
Display the `fbconf_xorg` command line options that are supported in conjunction with the frame buffer device, along with a brief explanation of each option. The frame buffer device can be specified using the `-dev` option.

`-prconf`
Display the current configuration for the frame buffer device and attached display device(s). The frame buffer device can be specified using the `-dev` option.

The `-prconf` output might resemble:

```
Monitor/Resolution Information:
    Monitor manufacturer: SUN
    Product Code:  4
    Serial Number: 12212555
    Manufacture date: 2000, week 9
    EDID Version:  1.1
    Monitor dimensions: 36x29 cm
    Default Gamma: 2.62
    Monitor preferred resolution: SUNW_STD_1280x1024x60
    Monitor supported resolutions from EDID: SUNW_STD_1280x1024x60,
        SUNW_STD_1280x1024x76, 1152x900x66, VESA_STD_1280x1024x75,
        VESA_STD_1280x1024x60, SUNW_STD_1152x900x66,
        VESA_STD_720x400x70, VESA_STD_640x480x60,
        VESA_STD_640x480x67, VESA_STD_640x480x72,
        VESA_STD_640x480x75, VESA_STD_800x600x56,
        VESA_STD_800x600x60, VESA_STD_800x600x72,
        VESA_STD_800x600x75, VESA_STD_832x624x75,
        VESA_STD_1024x768x60, VESA_STD_1024x768x70, 1024x768x75
    Current resolution setting: FALLBACK_1152x900x66

Framelock Configuration:
    Slave Mode: Disabled
```

`-predid` [`raw`] [`parsed`]

Display the E-EDID (Enhanced Extended Display Identification Data) information obtained from the display device(s), which must be online, connected to the frame buffer. The frame buffer device can be specified using the `-dev` option. The output is raw hexadecimal and/or human-readable (parsed) text. The default is parsed.

The `-predid` raw output might resemble:

```
--- EDID Data for /dev/fbs/kfb0 ---

Block 0: EDID Base Block
        0x00:  00 FF FF FF FF FF FF 00 04 43 06 F2 01 00 00 00
        0x10:  01 11 01 04 0F 2B 20 78 2B 9C 68 A0 57 4A 9B 26
        0x20:  12 48 4C FF FF 80 A9 59 A9 4F A9 4A A9 45 81 99
        0x30:  81 80 61 59 45 59 48 3F 40 30 62 B0 32 40 40 C0
        0x40:  13 00 AB 40 11 00 00 1E 00 00 00 FD 00 32 5A 1E
        0x50:  6E 17 04 11 00 C8 90 00 50 3C 00 00 00 F7 00 0A
        0x60:  F7 0F 03 87 C0 00 00 00 00 00 00 00 00 00 00 FC
        0x70:  00 41 42 43 20 4C 43 44 32 31 0A 20 20 20 00 0B
```

`-propt`

Display all option settings for the frame buffer device, either as they currently are or as they are represented in the `xorg.conf` configuration file when `fbconf_xorg` completes. The device can be specified using the `-dev` option, and the file using the `-file` option.

The `-propt` output might resemble:

```
--- Graphics Configuration for /dev/fbs/efb0 ---

xorg.conf: machine -- /etc/X11/xorg.conf
      Screen section:  "efb0"
      Device section:  "efb0"
      Monitor section: none

Video Mode: Not set

Screen Information:
      DoubleWide: Disable
      DoubleHigh: Disable
      Clone: Disable
      Offset/Overlap: [0, 0]
      Outputs:        Direct

Visual Information:
      Gamma Correction: Using default gamma value 2.22
```

`-res ?`

Display a list of video modes that can be used with the `-res` *video-mode* option.

The ? argument might need to be escaped or placed in quotes (\?, "?", or '?'), to protect it from misinterpretation by the shell.

The -res ? output might resemble:

```
Video modes accepted by the -res option:
     AUTO                      [1][2]
     NONE                      [1][2]
     SUNW_STD_1920x1200x75
     SUNW_STD_1920x1200x70
     SUNW_DIG_1920x1200x60
     SUNW_STD_1920x1080x72
     SUNW_DIG_1920x1080x60
       ...
       ...
     VESA_STD_640x480x75       [1]
     VESA_STD_640x480x72       [1]
     VESA_STD_640x480x60       [1]

[1] Resolution is supported by monitor
[2] Preferred resolution for monitor
```

Abbreviations such as "1280x1024x75" might also be used.

-res *video-mode* [nocheck | noconfirm]

Set the video mode for the display device that is connected to the frame buffer device.

A list of video modes can be displayed using the -res ? option.

The basic format of a video-mode is *width*x*height*x*rate*, where:

- *width* is the screen width in pixels.
- *height* is the screen height in pixels.
- *rate* is the vertical frequency of the screen refresh.

A video-mode argument might have an @ (at sign) instead of x preceding the refresh rate. For instance, 1280x1024x76 and 1280x1024@76 are equivalent.

A video-mode name might carry additional information, as with SUNW_STD_1280x1024x76.

The -res argument, auto, represents the video mode that is currently programmed into the device. The argument, none, is a synonym for auto.

Note that some video modes might be inappropriate for certain frame buffer devices and/or display devices.

The -res option accepts suboption keywords following the video-mode specification.

nocheck

The nocheck suboption causes the video-mode argument to be accepted, regardless of whether it is supported by the currently attached monitor, whether it is known within

the current configuration, and so forth. Note that using an unchecked, inappropriate video mode can leave the system without usable video output. This suboption is useful if a different monitor is to be connected to the frame buffer device. This suboption also implies `noconfirm`.

`noconfirm`
If the video-mode argument is unable to be validated, the default action is to display a warning message and ask the user whether to continue. The `noconfirm` suboption suppresses this confirmation request. This suboption is useful when `fbconf_xorg` is being run from a shell script.

The following device-specific options are supported for certain frame buffer devices. Unless specified otherwise, these options do not take effect until the user logs out and back in.

`-deflinear true | false`
This option selects the default X visual. Two types of visuals are supported, linear and nonlinear. Linear visuals are gamma corrected. Nonlinear visuals are not.

If the value of this option is `true`, the default visual is set to default depth 24 and the default class is `TrueColor` with gamma correction enabled. If `false`, a nonlinear visual that satisfies the other default visual selection options, such as the default depth and default class, is chosen as the default visual.

The `-deflinear`, `-defoverlay`, and `-deftransparent` options each select the default X visual. Only one of these might be enabled at a time. Enabling one causes the others to be disabled.

`-defoverlay true | false`
This option selects the default X visual. Some devices might provide an 8-bit PseudoColor visual whose pixels are disjoint from the rest of the visuals. This is called the overlay visual. Windows created in this visual does not damage windows created in other visuals. The converse, however, is not true: Windows created in other visuals damage overlay windows.

If the value of this option is `true`, the overlay visual is the default visual. The default depth is 8–bit and the default class is PseudoColor. If `false`, the non-overlay visual that satisfies the other default visual selection options, such as the default depth and the default class, is chosen as the default visual.

The `-deflinear`, `-defoverlay`, and `-deftransparent` options each select the default X visual. Only one of these might be enabled at a time. Enabling one causes the others to be disabled.

`-deftransparent true | false`
This option selects the default X visual. Some devices might provide an 8-bit PseudoColor visual whose pixels are disjoint from the rest of the visuals. This is called the overlay visual. Windows created in this visual does not damage windows created in other visuals.

If the value of this option is `true`, the overlay visual used as the default is a transparent overlay visual. A visual with transparency supports a colormap with 255 colors and one transparent pixel. The default depth is 8–bit and the default class is PseudoColor. If `false`, the nonoverlay visual that satisfies the other default visual selection options, such as the default depth and the default class, is chosen as the default visual.

The `-deflinear`, `-defoverlay`, and `-deftransparent` options each select the default X visual. Only one of these might be enabled at a time. Enabling one causes the others to be disabled.

`-doublehigh` `enable` | `disable`
Configures the two outputs of the frame buffer device into one vertical virtual display. The default is `disable`. The `-doublewide` and `-doublehigh` options are mutually exclusive. Enabling one causes the other to be disabled.

`-doublewide` `enable` | `disable`
Configures the two outputs of the frame buffer device into one horizontal virtual display. The default is `disable`. The `-doublewide` and `-doublehigh` options are mutually exclusive. Enabling one causes the other to be disabled.

`-g` *gamma-correction-value*
Sets the gamma correction value. All linear visuals provide gamma correction. The gamma correction value should be in the range, `0.1` to `10.0`. The default is `2.22`. This option can be used while the window system is running. Changing the gamma correction value affects all of the windows displayed by linear visuals.

`-gfile` *gamma-correction-file*
Loads the gamma correction table from the file specified by *gamma-correction-file*. This text file specifies the gamma correction values for the R, G, and B channels. Three consecutive values form an RGB triplet. For a `kfb` device, there must be exactly 256 RGB triplets. A value might be represented in hexadecimal, decimal, or octal format (for example, `0x3FF`, `1023`, or `01777`, respectively). Values are separated by one or more whitespace or new line characters. Comments begin with a hash sign character (#) and end at the end of the line.

You can load the gamma correction table with this option while the window system is running. The new gamma correction affects all the windows being displayed using the linear visuals. When gamma correction is done using a user-specified table, the gamma correction value (`-g`) is undefined. By default, the window system assumes a gamma correction value of `2.22` and loads the gamma table it creates corresponding to this value.

The following is an example of a *gamma-correction-file* file:

```
# Gamma Correction Table
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
 ...  ...  ...
```

```
                ...  ...  ...
               0xFF 0xFF 0xFF
```

`-multisample` `available` | `disable` | `forceon`
: If set to disable, no multisample is possible. If set to available, multisample is possible but is selected on a per-window basis using a library interface. If set to `forceon`, all Sun OpenGL windows are rendered using multisampling. To query the number of samples used, specify the `-propt` option.

`-offset` *x-val y-val*
: Adjusts the position of the specified stream by the value specified. This option is only implemented in `-doublewide` and `-doublehigh` modes. For `-doublewide`, use the *x-val* to position the rightmost stream. Negative is left (overlaps with the left stream). For `-doublehigh`, use the *y-val* to position the bottom stream. Negative is up (overlaps with top stream). The default is `[0,0]`.

`-samples` `1` | `2` | `4` | `8` | `16`
: Requests the number of samples to compute per display pixel. The requested number of samples per pixel is used if `-multisample` is not disabled and resources exist for the request. To query the number of samples used, specify the `-propt` option or run the `xglinfo` utility. The `xglinfo` utility can return the number of multisamples after you specify the option `-multisample` available. The default is `4`.

`-slave` `disable` | `multiview`
: If you set the `multiview` argument for the `-slave` option, the device synchronizes video with a master through the multiview genlock ribbon cable. The system should be powered off whenever connecting or disconnecting this cable. Both devices should be running the same resolution and the option should be issued when the window system is running. The default is `disable`.

Defaults

Certain options have implied default arguments. The default argument is used when the option is not present on the `fbconf_xorg` command line. For instance, a default argument for `-dev` is `/dev/fb`.

Options that set configuration state do not have implied defaults. The `-res` option is one example. If a configuration option is omitted from the `fbconf_xorg` command line, the corresponding `xorg.conf` configuration setting remains unchanged. The exception is that if configuration options are mutually exclusive, setting one automatically unsets each of the others. An example is `-deflinear`, `-defoverlay`, and `-deftransparent`.

If a configuration setting is not present in the configuration file when the window system is run, a default value is used. For instance, the default state associated with `-res` is `auto`. A setting might not be present in the file, or the file itself might not exist, until `fbconf_xorg` has been invoked with the corresponding command line option.

The `-defaults` option sets the default values for most configuration settings.

Options and their defaults are shown below.

Device-independent defaults:

```
Option          Default Argument
-dev            /dev/fb
-file           machine

Option          Default State
-res            auto
```

使用例

例 1 Switching the Resolution of the Monitor Type

The following example sets the video mode for the monitor on the `/dev/fbs/efb0` device to 1280 x 1024 at 76 Hz:

```
example% fbconf_xorg -dev efb0 -res 1280x1024x76
```

終了ステータス

The following exit values are returned:

`0`
: Execution completed successfully.

`1`
: Invalid command line usage.

`2`
: An error occurred.

ファイル

`/dev/fb`	Symbolic link to the default frame buffer device.
`/dev/fbs/efbn`*n*	Device special file for an `efb` frame buffer
`/usr/lib/fbconfig/SunModes_xorg.conf`	Video mode definitions included in new configuration files.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/graphics/fbconfig/fbconfig-kfb
Interface Stability	Uncommitted

関連項目

`fbconfig(1M)`, `svccfg(1M)`, `attributes(5)`, `efb(7D)`, `fbio(7I)`

See the Xorg(1) and Xserver(1) man pages in the X Server man page collection.

名前

fcinfo, fcadm – Fibre Channel HBA Port Command Line Interface

形式

```
fcinfo hba-port [-lite] [HBA_port_WWN]...

fcadm hba-port [-lite] [HBA_port_WWN]...

fcinfo remote-port [-ls] [-p HBA_port_WWN]
     [REMOTE_port_WWN]...

fcadm remote-port [-ls] [-p HBA_port_WWN]
     [REMOTE_port_WWN]...

fcinfo logical-unit | lu [-v] [OS device_path]

fcadm logical-unit | lu [-v] [OS device_path]

fcadm create-npiv-port -p Virtual_Port_WWN [-n Virtual_Node_WWN]
     PhysicalPort_port_WWN

fcadm delete-npiv-port -p Virtual_Port_WWN [-n Virtual_Node_WWN]
     PhysicalPort_port_WWN

fcadm create-fcoe-port [-tf] -p Port_WWN [-n Node_WWN]
     MAC_Interface

fcadm delete-fcoe-port MAC_Interface

fcadm list-fcoe-ports MAC_Interface

fcadm force-lip Port_WWN

fcinfo [-V]

fcadm [-V]

fcinfo [-?]

fcadm [-?]
```

機能説明

`fcinfo` and `fcadm` are command line interfaces that collect administrative information on fibre channel host bus adapter (HBA) ports on a host. They also collect data on any fibre channel targets that might be connected to those ports in a Storage Area Network (SAN).

A port can be either in initiator mode or target mode. The same FC attribute information and remote port information are applied to both the initiator and the target modes port except that SCSI-related operations described below do not apply for the target mode port.

SUBCOMMANDS

The following subcommands are supported by both `fcinfo` and `fcadm`:

`hba-port`
: Lists information for the HBA port referenced by the specified *HBA_port_WWN*. If *HBA_port_WWN* is not specified, all initiator mode and target mode fibre channel HBA ports on the host will be listed.

`remote-port` Lists the *remote-port* information for those remote ports that are specified. If no *REMOTE_port_WWN* is specified, all remote ports that are visible through *HBA_port_WWN* are listed.

`logical-unit` | `lu` Lists the logical unit referenced by the specified *device_path*. If *device_path* is not specified, all fibre channel logical units will be listed. This subcommand applies only to the initiator mode.

The following subcommands, which administer N_Port ID Virtualization (NPIV) HBA ports, are supported only by `fcadm`:

`create-npiv-port`
Create an NPIV virtual port on the specified adapter.

`delete-npiv-port`
Delete an NPIV virtual port. This delete only ports create by `fcadm`.

The following subcommands, which administer Fibre Channel over Ethernet (FCoE) ports, are supported only by `fcadm`:

`create-fcoe-port`
Creates a FCoE port associated with the given MAC interface and the given *Port_WWN*/*Node_WWN*. If `-p` or `-n` is not specified, the Port WWN or Node WWN will be generated automatically. If the specified MAC interface does not support multiple unicast address, no FCoE port will be created and you receive a message indicating that you can add an `-f` option to force enabling promiscuous mode on the specified MAC interface to enable FCoE.

`delete-fcoe-port`
Deletes the FCoE port associated with the specified MAC interface.

`list-fcoe-ports`
List information of FCoE ports.

`force-lip`
Force the link to reinitialize. When issuing this subcommand on the target port side, causes a reset of the target port. When issuing it from the host port side, resets the host port. When an FC switch is connected, other FC ports in the SAN will get a RSCN (Remote State Change Notification). Furthermore, other initiators will always rediscover the port after this, and the FC login session will be established or reused. Also, I/Os will be continued. This command is disruptive to I/Os, but it is not destructive, as it does not cause any data loss.

オプション

The following options are supported:

`-e`, `--fcoe`
Lists the information for all FCoE ports.

`-f`, `--fcoe-promiscuous`
Used with `create-fcoe-port` to create an FCoE port on a MAC interface associated with a NIC that does not support multiple unicast address. Promiscuous mode will be enabled for the specified MAC interface.

`-l`, `--linkstat`
Lists the link error statistics information for the port referenced by the specified *HBA_port_WWN* or *REMOTE_port_WWN*.

`-n` *HBA_node_WWN*, `--node` *HBA_node_WWN*
When used with NPIV options, specify a virtual node WWN. If used with `create-npiv-port`, it can be omitted, and a random based WWN will be used. It is mandatory for `delete-npiv-port`.

When used with `create-fcoe-port` subcommand, specify the node WWN for the FCoE port. It can be omitted, in which case a WWN will be generated based on the MAC address of the specified MAC interface.

`-p` *HBA_port_WWN*, `--port` *HBA_port_WWN*
Retrieve remote port information from the *HBA_port_WWN* of the local HBA port on the host. When used with the `remote-port` subcommand, the `-p` option is mandatory.

When used with NPIV options, specify a virtual port WWN. If used with `create-npiv-port`, it can be omitted, and a random based WWN will be used. It is mandatory for `delete-npiv-port`.

When used with `create-fcoe-port` subcommand, specify the port WWN for the FCoE port. It can be omitted, in which case a WWN will be generated based on the MAC address of the specified MAC interface.

`-s`, `--scsi-target`
Lists the SCSI target information for all remote ports the user has asked for. The `-p`, `--port` option must always be specified and must be a valid HBA port on the host. This HBA port will be used as the initiator for which to retrieve the SCSI level target information. Note that this will only function on remote port fibre channel World-Wide Names that support an FC4 type of SCSI. This option applies only to an initiator mode port.

`-t` [*HBA_node_WWN*], `-target` [*HBA_node_WWN*]
Lists the information for the port with the target mode referenced by the specified *HBA_node_WWN* or all target mode ports discovered in the host.

When used with `create-fcoe-port`, create a FCoE target mode port.

`-v`, `--verbose`
When used with the `logical-unit` subcommand, the `-v` displays additional information for the logical unit, including SCSI vendor and product information and device type as well as the FC World-Wide names for the local and remote fibre channel ports to which this device is attached.

`-V`, `--version`
Displays the version information.

`-?`, `--help`
Displays the usage information.

使用例

例 1 Listing All HBA Ports

The following command lists all initiator mode fibre channel HBA ports on the host:

```
# fcinfo hba-port

HBA Port WWN: 210000e08b074cb5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c1
        Manufacturer: QLogic Corp.
        Model: 375-3108-xx
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Serial Number: not available
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200000e08b074cb5
NPIV Port List:
        Virtual Port 1:
         Port WWN: 200000e08b074cb1
         Node WWN: 200000e08b074cb3
HBA Port WWN: 210100e08b274cb5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c2
        Manufacturer: QLogic Corp.
        Model: 375-3108-xx
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Serial Number: not available
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200100e08b274cb5
HBA Port WWN: 210000e08b072ab5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c3
```

例 1 Listing All HBA Ports (続き)

```
        Manufacturer: QLogic Corp.
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Model: 375-3108-xx
        Serial Number: not available
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: L-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200000e08b072ab5
HBA Port WWN: 210100e08b272ab5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c4
        Manufacturer: QLogic Corp.
        Model: 375-3108-xx
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Serial Number: 0402F00-0549112808
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200100e08b272ab5
```

例 2 Listing Target Mode HBA Ports

The following command lists all target mode fibre channel HBA ports on the host:

```
# fcinfo hba-port -t
  HBA Port WWN: 210100e08bb09221
     Port Mode: Target
     Port ID: 10700
     OS Device Name: Not Applicable
     Manufacturer: QLogic Corp.
     Model: d30ac7e0
     Firmware Version: 4.0.109
     FCode/BIOS Version: N/A
     Type: F-port
     State: online
     Supported Speeds: not established
     Current Speed: 2Gb
     Node WWN: 200100e08bb09221
  HBA Port WWN: 210000e08b909221
```

例 2 Listing Target Mode HBA Ports (続き)

```
    Port Mode: Target
    Port ID: 10900
    OS Device Name: Not Applicable
    Manufacturer: QLogic Corp.
    Model: d37ad1e0
    Firmware Version: 4.0.109
    FCode/BIOS Version: N/A
    Type: F-port
    State: online
    Supported Speeds: not established
    Current Speed: 2Gb
    Node WWN: 200000e08b909221
HBA Port WWN: 200000144fc2d508
    Port Mode: Target
    Port ID: 9a0025
    OS Device Name: Not Applicable
    Manufacturer: Sun Microsystems, Inc.
    Model: FCoE Virtual FC HBA
    Firmware Version: N/A
    FCode/BIOS Version: N/A
    Serial Number: N/A
    Driver Name: COMSTAR FCOET
    Driver Version: 1.0
    Type: F-port
    State: online
    Supported Speeds: 1Gb 10Gb
    Current Speed: 10Gb
    Node WWN: 100000144fc2d508
HBA Port WWN: 200000144fc2d509
    Port Mode: Target
    Port ID: 9a0023
    OS Device Name: Not Applicable
    Manufacturer: Sun Microsystems, Inc.
    Model: FCoE Virtual FC HBA
    Firmware Version: N/A
    FCode/BIOS Version: N/A
    Serial Number: N/A
    Driver Name: COMSTAR FCOET
    Driver Version: 1.0
    Type: F-port
    State: online
    Supported Speeds: 1Gb 10Gb
    Current Speed: 10Gb
    Node WWN: 100000144fc2d509
```

例 3 Listing HBA Ports and Link Statistics

The following command lists information for the HBA ports and the link statistics for those ports:

```
# fcinfo hba-port -l 210000e08b074cb5 210100e08b274cb5

HBA Port WWN: 210000e08b074cb5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c1
        Manufacturer: QLogic Corp.
        Model: 375-3108-xx
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Serial Number: 0402F00-0549112808
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200000e08b074cb5
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
HBA Port WWN: 210100e08b274cb5
        Port Mode: Initiator
        OS Device Name: /dev/cfg/c2
        Manufacturer: QLogic Corp.
        Model: 375-3108-xx
        Firmware Version: 3.3.116
        FCode/BIOS Version: 1.13.08
        Serial Number: 0402F00-0549112808
        Driver Name: qlc
        Driver Version: 20070212-2.19
        Type: N-port
        State: online
        Supported Speeds: 1Gb 2Gb
        Current Speed: 2Gb
        Node WWN: 200100e08b274cb5
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
```

例 3 Listing HBA Ports and Link Statistics (続き)

```
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
```

例 4 Listing All Remote Ports

The following command lists all remote ports that are visible through the given HBA port:

```
# fcinfo remote-port -p 210100e08b274cb5

Remote Port WWN: 50020f230000b4af
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 50020f200000b4af
Remote Port WWN: 210000e08b07daa6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200000e08b07daa6
Remote Port WWN: 20030003ba27c788
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 10000003ba27c788
Remote Port WWN: 210000e08b096a60
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200000e08b096a60
```

例 5 Listing Remote Ports and Link Statistics

The following command lists information for the remote ports and the link statistics for those ports:

```
# fcinfo remote-port -l -p 210100e08b272ab5

Remote Port WWN: 210100e08b296a60
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296a60
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
```

例 5 Listing Remote Ports and Link Statistics (続き)

```
                Invalid CRC Count: 0
Remote Port WWN: 20030003ba27d56d
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 10000003ba27d56d
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 4765165
                Loss of Signal Count: 4765165
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210100e08b27f7a6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27f7a6
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 50020f230000b897
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 50020f200000b897
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 7
                Loss of Signal Count: 7
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210100e08b27daa6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27daa6
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
```

例 5 Listing Remote Ports and Link Statistics (続き)

```
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210000e08b074cb5
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200000e08b074cb5
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210100e08b296060
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296060
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
```

例 6 Listing All SCSI Targets and Link Statistics

The following command lists all remote ports as well as the link statistics and *scsi-target* information:

```
# fcinfo remote-port  -sl -p 210100e08b272ab5

Remote Port WWN: 210100e08b296a60
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296a60
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
```

例 6 Listing All SCSI Targets and Link Statistics (続き)

```
Remote Port WWN: 20030003ba27d56d
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 10000003ba27d56d
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 4765165
                Loss of Signal Count: 4765165
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
        LUN: 0
          Vendor: SUN
          Product: T4
          OS Device Name: /dev/rdsk/c4t20030003BA27D56Dd0s2
        LUN: 1
          Vendor: SUN
          Product: T4
          OS Device Name: /dev/rdsk/c4t20030003BA27D56Dd1s2
Remote Port WWN: 210100e08b27f7a6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27f7a6
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 50020f230000b897
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 50020f200000b897
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 7
                Loss of Signal Count: 7
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
        LUN: 0
          Vendor: SUN
```

例 6 Listing All SCSI Targets and Link Statistics (続き)

```
    Product: T300
    OS Device Name: Unknown
  LUN: 1
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d1s2
  LUN: 2
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d2s2
  LUN: 3
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d3s2
  LUN: 4
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d4s2
  LUN: 5
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d5s2
  LUN: 6
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d6s2
  LUN: 7
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d7s2
  LUN: 8
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d8s2
  LUN: 9
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d9s2
  LUN: 10
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d10s2
  LUN: 11
    Vendor: SUN
    Product: T300
    OS Device Name: /dev/rdsk/c4t50020F230000B897d11s2
```

例 6 Listing All SCSI Targets and Link Statistics (続き)

```
        LUN: 12
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d12s2
        LUN: 13
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d13s2
        LUN: 14
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d14s2
        LUN: 15
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d15s2
Remote Port WWN: 210100e08b27daa6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27daa6
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210000e08b074cb5
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200000e08b074cb5
        Link Error Statistics:
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
Remote Port WWN: 210100e08b296060
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296060
        Link Error Statistics:
```

例 6 Listing All SCSI Targets and Link Statistics (続き)

```
                Link Failure Count: 0
                Loss of Sync Count: 0
                Loss of Signal Count: 0
                Primitive Seq Protocol Error Count: 0
                Invalid Tx Word Count: 0
                Invalid CRC Count: 0
```

例 7 Listing SCSI Target Information

The following command lists all remote ports as well as the *scsi-target* information:

```
# fcinfo remote-port  -s -p 210100e08b272ab5

Remote Port WWN: 210100e08b296a60
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296060
Remote Port WWN: 20030003ba27d56d
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 10000003ba27d56d
        LUN: 0
          Vendor: SUN
          Product: T4
          OS Device Name: /dev/rdsk/c4t20030003BA27D56Dd0s2
        LUN: 1
          Vendor: SUN
          Product: T4
          OS Device Name: /dev/rdsk/c4t20030003BA27D56Dd1s2
Remote Port WWN: 210100e08b27f7a6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27f7a6
Remote Port WWN: 50020f230000b897
        Active FC4 Types: SCSI
        SCSI Target: yes
        Port Symbolic Name: unknown
        Node WWN: 50020f200000b897
        LUN: 0
          Vendor: SUN
          Product: T300
          OS Device Name: Unknown
        LUN: 1
          Vendor: SUN
```

例 7 Listing SCSI Target Information (続き)

```
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d1s2
LUN: 2
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d2s2
LUN: 3
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d3s2
LUN: 4
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d4s2
LUN: 5
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d5s2
LUN: 6
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d6s2
LUN: 7
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d7s2
LUN: 8
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d8s2
LUN: 9
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d9s2
LUN: 10
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d10s2
LUN: 11
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d11s2
LUN: 12
  Vendor: SUN
  Product: T300
  OS Device Name: /dev/rdsk/c4t50020F230000B897d12s2
```

例 7 Listing SCSI Target Information (続き)

```
        LUN: 13
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d13s2
        LUN: 14
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d14s2
        LUN: 15
          Vendor: SUN
          Product: T300
          OS Device Name: /dev/rdsk/c4t50020F230000B897d15s2
Remote Port WWN: 210100e08b27daa6
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b27daa6
Remote Port WWN: 210000e08b074cb5
        Active FC4 Types: SCSI,IP
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200000e08b074cb5
Remote Port WWN: 210100e08b296060
        Active FC4 Types: SCSI
        SCSI Target: no
        Port Symbolic Name: unknown
        Node WWN: 200100e08b296060
```

例 8 Listing the Logical Unit

The following command lists the logical unit:

```
# fcinfo logical-unit

/dev/rdsk/c0t600C0FF0000000000036223AE73EB705d0s2
/dev/rdsk/c0t600C0FF0000000000000362272539E5B03d0s2
/dev/rmt/0n
```

例 9 Displaying Additional Information for the Logical Unit

The following command displays additional information about the logical unit using the `-v` option for device `/dev/rmt/0n`:

```
# fcinfo lu -v /dev/rmt/0n

 OS Device Name: /dev/rmt/0n
        HBA Port WWN: 210000e07c030b91
                  Remote Port WWN: 21010003b7830a6
```

例 9 Displaying Additional Information for the Logical Unit (続き)

```
                        LUN: 0
        Vendor: STK
        Product: 9940A
        Device Type: Tape device
```

The following command displays additional information about the logical unit using the `-v` option for device `/dev/rdsk/c0t600C0FF0000000000036223AE73EB705d0s2`

```
# fcinfo logical-unit -v \
     /dev/rdsk/c0t600C0FF0000000000036223AE73EB705d0s2

 OS Device Name: /dev/rdsk/c0t600C0FF0000000000036223AE73EB705d0s2
        HBA Port WWN: 210100e08b27a8a1
                Remote Port WWN: 256000c0ffc03622
                        LUN: 0
                Remote Port WWN: 216000c0ff803622
                        LUN: 0
        HBA Port WWN: 210000e08b07a8a1
                Remote Port WWN: 256000c0ffc03622
                        LUN: 0
                Remote Port WWN: 216000c0ff803622
                        LUN: 0
        Vendor: SUN
        Product: StorEdge 3510
        Device Type: Disk device
```

例 10 Adding an NIPV Port

The following command adds an NPIV port to the HBA with a port name:

```
210000e08b170f1c
```

The NPIV port has a port name of `2000000000000001` and a node name of `2100000000000001`.

```
# fcadm create-npiv-port  -p 2000000000000001 -n 2100000000000001 \
210000e08b170f1c

Created NPIV Port:
    HBA Port WWN: 2000000000000001
    Node WWN: 2100000000000001
    Physical HBA Port WWN: 210000e08b170f1c
```

例 11 Adding an NIPV Port with Random WWN

The following command adds an NPIV port to the HBA with a randomly assigned port name of `210000e08b170f1c`.

例 11 Adding an NIPV Port with Random WWN *(続き)*

```
# fcadm create-npiv-port 210000e08b170f1c

Created NPIV Port:
    HBA Port WWN: 2038295824942801
    Node WWN: 2100392849248001
    Physical HBA Port WWN: 210000e08b170f1c
```

例 12 Deleting an NIPV Port

The following command deletes an NPIV port.

```
# fcadm delete-npiv-port -p 2000000000000001 -w 2100000000000001 \
210000e08b170f1c
```

例 13 Creating an FCoE Target Port

The following command creates an FCoE port associated with the MAC interface `nxge0`.

```
# fcadm create-fcoe-port -t nxge0
```

例 14 Deleting an FCoE Port

The following command deletes the FCoE port associated with the MAC interface `nxge0`.

```
# fcadm delete-fcoe-port -t nxge0
```

例 15 Listing Information for an FCoE Port

The following command lists information for FCoE ports.

```
# fcadm list-fcoe-ports
HBA Port WWN: 200000144fc2d508
        Port Type: Target
        MAC Name: nxge0
        MTU Size: 9194
        MAC Factory Address: 00144fc2d508
        MAC Current Address: 00144fc2d508
HBA Port WWN: 200000144fc2d509
        Port Type: Target
        MAC Name: nxge1
        MTU Size: 9194
        MAC Factory Address: 00144fc2d509
        MAC Current Address: 00144fc2d509
```

例 16 Reinitializing the Link of an FC Port

The following command forces the link connected with the port `200000144fc2d508` to reinitialize.

```
# fcadm force-lip 200000144fc2d508
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/storage/fc-utilities
Interface Stability	Committed

関連項目

attributes(5)

Error Messages

Errors that can occur in addition to the errors normally associated with system administration commands:

- *HBA_port_WWN*: `not found`
- *Remote_port_WWN*: `not found`
- *HBA_port_WWN*: `NPIV not supported on this HBA`

名前

fdetach – detach a name from a STREAMS-based file descriptor

形式

`fdetach` *path*

機能説明

The `fdetach` command detaches a STREAMS-based file descriptor from a name in the file system. Use the *path* operand to specify the path name of the object in the file system name space, which was previously attached. See `fattach(3C)`.

The user must be the owner of the file or a user with the appropriate privileges. All subsequent operations on *path* will operate on the underlying file system entry and not on the STREAMS file. The permissions and status of the entry are restored to the state they were in before the STREAMS file was attached to the entry.

オペランド

The following operands are supported:

path Specifies the path name of the object in the file system name space, which was previously attached.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fattach(3C)`, `fdetach(3C)`, `attributes(5)`, `streamio(7I)`

『STREAMS Programming Guide』

名前

fdisk – 固定ディスクパーティションテーブルの作成または変更

形式

```
fdisk [-o offset] [-s size] [-P fill_patt] [-S geom_file]
      [-w | -r | -d | -n | -I | -B | -t | -T | -g | -G | -R | -E]
      [--F fdisk_file] [ [-v] -W {fdisk_file | –}]
      [-h] [-b masterboot]
      [-A id : act : bhead : bsect : bcyl : ehead : esect :
          ecyl : rsect : numsect]
      [-D id : act : bhead: bsect : bcyl : ehead: esect :
          ecyl : rsect : numsect] rdevice
```

機能説明

このコマンドは、次のように使用します。

- x86 システム上の `fdisk` パーティションテーブルを作成および変更する
- SPARC または x86 システム上のリムーバブルメディアで `fdisk` パーティションテーブルを作成および変更する
- x86 システムのみ、固定ディスクの最初のセクターに格納されるマスターブートレコードをインストールする

このテーブルは、異なるオペレーティングシステムごとに予約されているディスク領域を識別するために、または第 2 段階のブートストラップのあるパーティション (アクティブな Solaris パーティション) を識別するために、第 1 段階のブートストラップ (またはファームウェア) により使用されます。*rdevice* 引数は、固定ディスクに対応する raw デバイスを指定する場合に使用しなくてはいけません。たとえば、`/dev/rdsk/c0t0d0p0` のように指定します。

このプログラムは、3 種類の動作モードで実行します。最初の動作モードは対話モードです。対話モードでは、ディスク上に存在するパーティションテーブルが表示され、さらにユーザーがテーブルを変更できるように、メニューが表示されます。メニュー、問い合わせ、警告、およびエラーメッセージは、特に説明がなくても理解できるものです。

対話モードでは、ディスク上にパーティションテーブルがない場合、ユーザーはデフォルトのパーティション分割を行うか、または初期テーブル値を指定するかの、どちらかを選択できます。デフォルトのパーティション分割では、ディスク全体が Solaris システムに割り当てられ、Solaris システムパーティションがアクティブになります。いずれの場合も、初期テーブルが作成された時点で、`fdisk` はパーティションテーブルとともに、第 1 段階のブートストラップ (x86 のみ) コードも書き込みます。このモード (x86 のみ) では、2T バイトより大きなディスク上に非 EFI パーティションのエントリを作成する場合、`fdisk` はパーティションの最大サイズが 2T バイトであることを警告します。これらの状態で、`fdisk` によって表示される割合 (パーセント) は 2T バイトに基づいています。

2 番目の動作モードは、エントリの追加、削除、または `fdisk` テーブル全体の置き換えを自動的に行う場合に使用します。このモードでは、コマンド行に指定したエントリを追加または削除できます。また、ファイルから `fdisk` テーブル全体を読み込んで、元のテーブルと置き換えることもできます。`fdisk` はこのファイルを作成

する目的でも使用できます。コマンド行オプションを使用すると、任意の fdisk テーブルが、ディスク全体を Solaris システムに割り当てるデフォルトの fdisk テーブルに置き換えられます。

3 番目の動作モードは、ディスクの診断時に使用します。このモードでは、ディスクのあるセクションをユーザーが指定したパターンで埋めることができ、ディスクのモードセクションを読み書きすることもできます。

注 – 3 番目の動作モードは、拡張パーティションでは現在サポートされません。

fdisk がパーティションを作成する場合、領域は fdisk のパーティションテーブル内に割り当てられますが、割り当てられたディスク領域は初期化されません。ファイルシステムのメタデータを作成して新しいパーティションに書き込むには newfs(1M) が必要であり、VTOC または EFI/GPT メタデータを書き込むには format(1M) が必要です。

メニューオプション

fdisk プログラムの対話モードで使用できるメニューオプションは、次のとおりです。

`Create a partition`
: このオプションを使用すると、新しいパーティションを作成できます。パーティションの最大数は 4 つです。パーティションのタイプ (SOLARIS、MS-DOS、UNIX、またはその他) が尋ねられます。次に、パーティションのサイズをディスクの割合で指定するように求められます。この時点で c と入力すると、開始シリンダ番号およびシリンダ内のパーティションのサイズが尋ねられます。c を入力しなかった場合は、パーティションが収まるように、fdisk プログラムが開始シリンダ番号を決定します。いずれの場合でも、パーティションが既存のパーティションとオーバーラップする場合、または収まりきらない場合は、メッセージが表示されて最初のメニューに戻ります。

`Change Active (Boot from) partition`
: このオプションを使用すると、第 1 段階のブートストラップが第 2 段階のブートストラップを検索するパーティション (アクティブパーティションと呼ばれる) を指定できます。

`Delete a partition`
: このオプションを使用すると、作成済みのパーティションを削除できます。そのパーティション内のすべてのデータが破壊されるので注意してください。

`Change between Solaris and Solaris2 Partition IDs`
: このオプションを使用すると、ユーザーは、現在と以前の fdisk オペレーティングシステムパーティション ID を切り替えることができます。これにより、そのディスクパーティション内のデータが影響を受けることはありません。このオプションは、以前のソフトウェアとの互換性を維持するために提供されています。

`Edit/View extended partitions`

このオプションを使用すると、拡張パーティションメニューを表示できます。論理ドライブの追加および削除、論理ドライブの sysid の変更、および論理ドライブ情報の表示を行うには、拡張パーティションメニューを使用します。拡張パーティションに加えた変更を確定するには、拡張パーティションサブメニューのオプション `r` を使用してメインメニューに戻る必要があります。拡張パーティションサブメニューによってサポートされるオプションのリストを表示するオプションもあります。オプションのリストを次に示します。

`a` 論理ドライブを追加します。

論理ドライブを追加するには、このサブメニューオプションを使用します。開始シリンダ、サイズ (シリンダ単位、または人間が読み取れる形式 - KB、MB、または GB)、およびパーティション ID の 3 種類の情報が必要となります。パーティション ID を指定する場合、サポートされるパーティションを一覧表示するために使用できるオプション (`I`) があります。

`d` 論理ドライブを削除します。

論理ドライブを削除するには、このサブメニューオプションを使用します。必要となる入力は、削除する論理ドライブの番号のみです。

`h` ヘルプメニューを表示します。

このサブメニューオプションは、拡張パーティションサブメニューでサポートされる操作を表示します。

`i` 論理ドライブの ID を変更します。

既存の論理ドライブのシステム ID を変更するには、このサブメニューオプションを使用します。このサブメニューで `I` オプションを使用すると、サポートされるシステム ID のリストが表示されます。

`p` 論理ドライブのレイアウトを表示します。

論理ドライブの情報を標準出力に表示します。この出力には、現在の `fdisk` プログラムの実行で加えられた変更が反映されます。変更がディスクに確定されるのは、メインメニューに戻って (サブメニュー `r` を使用)、変更をディスクに確定するオプションを選択したときです。

`r` メイン `fdisk` メニューに戻ります。

拡張パーティションサブメニューを終了して、メインメニューに戻ります。

拡張パーティションの番号付けの動的な特性に注意してください。たとえば、Solaris システムにパーティション `p1`、`p2`、`p3`、および `p4` があるとしま

す。拡張パーティションの作成のあと、同じシステムに論理デバイスノード p5 および以降のノードが最大 p36 まで連番で作成されました。たとえば、論理ドライブ p8 が削除されると、p8 に続くすべてのノード (p9 から p36 まで) がパーティションのリスト内で 1 つ上に移動し、p9 が p8 に、p10 が p9 というようになります。

この時点で、次に示すオプションを使用してパーティションテーブルの設定を変更することも、あるいは、テーブルを変更せずにセッションを中止することもできます。

Exit　このオプションを使用すると、fdisk によってこのセッションで作成された新しいテーブルが固定ディスクに書き込まれ、fdisk は終了します。

Cancel　パーティションテーブルを変更せずに終了します。

オプション

fdisk で使用できるオプションは、次のとおりです。

-A *id:act:bhead:bsect:bcyl:ehead:esect:ecyl:rsect:numsect*
引数で指定されたパーティションを追加します (形式については以下の -F オプションを参照)。fdisk テーブルが変更された場合、このオプションを使用すると、Solaris パーティション上の VTOC がゼロで埋められます。

-b *master_boot*
マスターブートプログラムとしてファイル *master_boot* を指定します。デフォルトのマスターブートプログラムは /usr/lib/fs/ufs/mboot です。

-B
デフォルトでディスク全体を 1 つの Solaris パーティションに割り当てます。x86 マシンでは、ディスクが 2T バイトより大きい場合、Solaris パーティションのデフォルトのサイズは 2T バイトに制限されます。

-d
詳細デバッグモードを有効にします。fdisk を使用するたびに、その状態を標準エラー出力に送ります。このオプションの出力を -F オプションで使用しないでください。

-D *id:act:bhead:bsect:bcyl:ehead:esect:ecyl:rsect:numsect*
引数で指定されたパーティションを削除します (形式については以下の -F オプションを参照)。引数が完全に一致しないと、エントリが削除されないことに注意してください。fdisk テーブルが変更された場合、このオプションを使用すると、Solaris パーティション上の VTOC がゼロで埋められます。

-E
ディスク全体を使用する EFI パーティションを作成します。

-F *fdisk_file*
ファイル *fdisk_file* を使用してテーブルを初期化します。fdisk テーブルが変更された場合、このオプションを使用すると、Solaris パーティション上の VTOC がゼロで埋められます。

fdisk_file には、一次パーティション用の 4 つの指定行とそれに続く論理ドライブ用の指定行が含まれています。少なくとも 1 つの論理ドライブがある場合、一次パーティションに対して 4 つの行がある必要があります。この場合、構成する一次パーティションの数が 4 つより少ないときは、残りの行をゼロで埋める必要があります。

各行は位置に依存するエントリからなり、エントリはスペースまたはコロンで区切ります。形式は次のとおりです。

id act bhead bsect bcyl ehead esect ecyl rsect numsect

エントリの値は次のとおりです。

id パーティションのタイプ。指定可能な数値は、`fdisk.h` に記述されています。

act アクティブパーティションプラグ。`0` は非アクティブ、`128` はアクティブを意味します。論理ドライブの場合、ユーザーが 128 を指定しても、このフラグは常に 0 に設定されます。

bhead パーティションが開始するヘッド。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

bsect パーティションが開始するセクター。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

bcyl パーティションが開始するシリンダ。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

ehead パーティションが終了するヘッド。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

esect パーティションが終了するセクター。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

ecyl パーティションが終了するシリンダ。`0` に設定されている場合、`fdisk` は他の情報に基づいて適切な値を設定します。

rsect パーティションが開始するディスクの先頭からの相対的な位置にあるセクター。この指定は必須です。この値は、`fdisk` が他のフィールドを設定するのに使用できます。論理ドライブの場合、論理ドライブに指定された *rsect* の前に、空きセクターが少なくとも 63 個ある必要があります。

numsect セクター数で表したこのディスクパーティションのサイズ。この指定は必須です。この値は、`fdisk` が他のフィールドを設定するのに使用できます。

-g
ディスクのラベルジオメトリを取得して、標準出力に出力します (形式については -S オプションを参照)。

-G
ディスクの物理ジオメトリを取得して、標準出力に出力します (形式については -S オプションを参照)。

-h
詳細メッセージを表示します。メッセージには全オプションのリストとともに、各オプションの説明が示されます。

-I
デバイスチェックを省略します。デバイスを使用せずに、ディスクに送られる内容のファイルイメージを生成する場合に使用します。このオプションは -S と組み合わせて使用する必要があります (前述を参照)。

-n
他のオプションで明示的に指定されている場合を除き、fdisk テーブルを更新しません。他のオプションを指定しない場合、-n オプションは、マスターブートレコードだけをディスクに書き込みます。また、-n オプションを指定した場合、fdisk は対話モードで起動しないことにも注意してください。

-o *offset*
ディスクの先頭からのブロックオフセット。このオプションは -P、-r、および -w とともに使用します。このオプションを指定しない場合、ゼロが想定されます。

-P *fill_patt*
パターン *fill_patt* でディスクを埋めます。*fill_patt* は 10 進数または 16 進数にすることができ、一定のロングパターンを表す数として使用します。*fill_patt* が # の場合、パターンは各ブロックで # になります。パターンはロングワードとして各ブロックに格納されて各ブロックを埋めます (-o および -s を参照)。

-r
ディスクから読み込んで標準出力に書き込みます。操作の開始ポイントおよびサイズを指定するオプション -o と -s を参照してください。

-R
読み取り専用ディスクとして扱います。このオプションはテスト用です。

-s *size*
操作を実行するブロック数 (-oを参照)

-S *geom_file*
geom_file の内容をラベルジオメトリとして設定します。*geom_file* には、行ごとに 1 つの指定を記述します。各行は復帰改行文字 (\n) で区切ります。行の先頭文字がアスタリスク (*) の場合、その行はコメント行とみなされます。各行は位置に依存するエントリからなり、スペースまたはコロンで区切ります。形式は次のとおりです。

pcyl ncyl acyl bcyl nheads nsectors sectsiz

エントリの値は次のとおりです。

pcyl ドライブの物理シリンダ数

ncyl ドライブの使用可能シリンダ数

acyl ドライブの代替シリンダ数

bcyl ドライブのオフセットシリンダ数 (ゼロにすべき)

nheads このドライブのヘッド数

nsectors トラックあたりのセクター数

sectsiz セクターサイズ (バイト単位)

`-t`
パーティションテーブル境界を越えないように、無効なスライステーブルエントリを調整します。

`-T`
パーティションテーブル境界にまたがる無効なスライステーブルエントリを削除します。

`-v`
HBA (仮想) ジオメトリディメンションを出力します。このオプションは `-W` フラグと組み合わせて使用する必要があります。このオプションは、仮想ジオメトリをサポートするプラットフォームで使用できます。(x86 のみ) 。

`-w`
ディスクに書き込み、標準入力から読み込みます。操作の開始ポイントおよびサイズを指定するオプション `-o` と `-s` を参照してください。

`-W -`
ディスクテーブルを標準出力に書き込みます。

`-W` *fdisk_file*
ディスクテーブルから `fdisk` ファイル *fdisk_file* を作成します。これは、上記の `-F` オプションと組み合わせて使用できます。

ファイル

`/dev/rdsk/c0t0d0p0` 固定ディスクに対応づけられる raw デバイス

`/usr/lib/fs/ufs/mboot` デフォルトのマスターブートプログラム

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
アーキテクチャ	x86 と SPARC

属性タイプ	属性値
使用条件	system/core-os

関連項目

uname(1), fmthard(1M), format(1M), newfs(1M), parted(1M), prtvtoc(1M), attributes(5)

診断

ほとんどのメッセージは、読めば理解できるものです。プログラムの開始直後に次のメッセージが表示される場合があります。

`Fdisk: cannot open <device>`
: このメッセージは、デバイス名引数が無効であることを示します。

`Fdisk: unable to get device parameters for device <device>`
: このメッセージは、固定ディスクの構成に問題があるか、または固定ディスクドライバでエラーが発生したことを示します。

`Fdisk: error reading partition table`
: このメッセージは、固定ディスクから最初の読み込み時にエラーが発生したことを示します。固定ディスクのコントローラまたはドライバに問題があるか、固定ディスクの構成に問題がある可能性があります。

`Fdisk: error writing boot record`
: このメッセージは、固定ディスクに新しいパーティションテーブルを書き込むときに、エラーが発生したことを示します。固定ディスクコントローラ、ディスク自体、ドライバ、または固定ディスクの構成に問題がある可能性があります。

名前

ff – list file names and statistics for a file system

形式

`ff [-F` *FSType*`] [-V] [`*generic_options*`] [-o` *specific_options*`]` *special*`...`

機能説明

`ff` prints the pathnames and inode numbers of files in the file system which resides on the special device *special*. Other information about the files may be printed using options described below. Selection criteria may be used to instruct `ff` to only print information for certain files. If no selection criteria are specified, information for all files considered will be printed (the default); the `-i` option may be used to limit files to those whose inodes are specified.

Output is sorted in ascending inode number order. The default line produced by `ff` is:

path-name i-number

The maximum information the command will provide is:

path-name i-number size uid

オプション

`-F` Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by matching the *special* with an entry in the table, or by consulting `/etc/default/fs`.

`-V` Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from `/etc/vfstab`. This option may be used to verify and validate the command line.

generic_options Options that are supported by most FSType-specific modules of the command. The following options are available:

- `-I` Do not print the i-node number after each path name.
- `-l` Generate a supplementary list of all path names for multiply-linked files.
- `-p` *prefix* The specified *prefix* will be added to each generated path name. The default is '.' (dot).
- `-s` Print the file size, in bytes, after each path name.
- `-u` Print the owner's login name after each path name.
- `-a -`*n* Select if the file has been accessed in *n* days.
- `-m -`*n* Select if the file has been written or created in *n* days.
- `-c -`*n* Select if file's status has been changed in *n* days.
- `-n` *file* Select if the file has been modified more recently than the argument `file`.

`-i` *i-node-list* Generate names for only those i-nodes specified in *i-node-list*. *i-node-list* is a list of numbers separated by commas (with no intervening spaces).

`-o` Specify *FSType*-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the *FSType*-specific module of the command.

オペランド

special A special device.

使用法

See `largefile(5)` for the description of the behavior of `ff` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/default/fs` default local file system type. Default values can be set for the following flags in `/etc/default/fs`. For example: `LOCAL=ufs`

`LOCAL` The default partition for a command if no FSType is specified.

`/etc/vfstab` list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`find(1)`, `ncheck(1M)`, `stat(2)`, `vfstab(4)`, `attributes(5)`, `largefile(5)` Manual pages for the FSType-specific modules of `ff`.

注意事項

This command may not be supported for all *FSTypes*.

The `-a`, `-m`, and `-c` flags examine the `st_atime`, `st_mtime`, and `st_ctime` fields of the `stat` structure respectively. (See `stat(2)`.)

名前

ff_ufs – list file names and statistics for a ufs file system

形式

`ff -F ufs [generic_options] [-o a,m,s] special...`

機能説明

`ff` prints the pathnames and inode numbers of files in the file system which resides on the special device *special*.

See `ff(1M)` for information regarding the `ff` command. See `OPTIONS` for information regarding the ufs-specific options.

オプション

The following options are supported:

`-o` Specify `ufs` file system specific options. The following options available are:

- `a` Print the '.' and '. .' directory entries.
- `m` Print mode information. This option must be specified in conjunction with the `-i` *i-node-list* option (see `ff(1M)`).
- `s` Print only special files and files with set-user-ID mode.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`find(1)`, `ff(1M)`, `ncheck(1M)`, `attributes(5)`

名前

fiocompress – file compression utility

形式

`/usr/sbin/fiocompress -c [-m] [-b` *block_size*`]` *input_file output_file*

`/usr/sbin/fiocompress -d` *input_file output_file*

機能説明

The `fiocompress` utility is a file compression tool that works together with the `dcfs(7FS)` file system to perform per-file compression. You can use `fiocompress` to decompress a compressed file or mark a compressed file as compressed, causing automatic decompression on read. The primary use of `fiocompress` is to compress files in the boot archive.

Note that this utility is not a Committed interface. See `attributes(5)`.

オプション

The following options are supported:

`-b` *block_size*
: Specify a block size for compression. The default block size is 8192.

`-c`
: Compress the specified file.

`-d`
: Decompress the specified file.

`-m`
: Mark the compressed file for automatic decompression on read. Can be used only in conjunction with `-c`.

終了ステータス

`0`
: The command completed successfully.

`–1`
: The command exited due to an error.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Private

関連項目

`boot(1M)`, `bootadm(1M)`, `dcfs(7FS)`, `ufs(7FS)`, `attributes(5)`

注意事項

This compression/decompression utility works only with files stored in a UFS file system.

There is no obvious way to determine whether a given file is compressed, other than copying the file and comparing the number of disk blocks of the copy against the original.

名前

flowadm – プロトコル、サービス、コンテナ、および仮想マシンの帯域幅リソース制御の管理

形式

```
flowadm show-flow [-P] [[-p] -o field[,...]] [-l link] [flow]

flowadm add-flow [-t] [-R root-dir] -l link -a attr=value[,...]
     [-p prop=value[,...]] flow
flowadm remove-flow [-t] [-R root-dir] {-l link | flow}

flowadm set-flowprop [-t] [-R root-dir] -p prop=value[,...] flow
flowadm reset-flowprop [-t] [-R root-dir] [-p prop[,...]] flow
flowadm show-flowprop [-P] [[-c] -o field[,...]] [-l link]
     [-p prop[,...]] [flow]
```

機能説明

flowadm コマンドは、特定リンク上のあるトラフィックタイプに対するネットワーク帯域幅とその関連リソースを作成、変更、削除、および表示するために使用されます。

flowadm コマンドを使用すると、トランスポート、サービス、またはサブネットのネットワーク帯域幅リソースを管理できます。サービスは、トランスポートとローカルポートを組み合わせて指定します。サブネットは、その IP アドレスとサブネットマスクで指定します。このコマンドは、物理リンク、仮想 NIC、リンク集積体など、任意のタイプのデータリンクで使用できます。

フローは、レイヤー 3 およびレイヤー 4 のヘッダーに基づく一連の属性として定義され、プロトコル、サービス、または仮想マシンを識別する目的で使用できます。あるフローがフロー属性に基づいて識別されると、そのフローに対するほかのトラフィックからの影響が最小限またはゼロに抑えられるように、そのフローに対してレイヤー 2、3、および 4 のキューやその処理用スレッド、その他のリソースなど、個別のカーネルリソースが一意に作成されます。

パフォーマンスへの影響を最小限に抑えながら制限を適用できるように、非常に高速かつスケーラビリティーの高い方法でインバウンドおよびアウトバウンドのパケットがフローに対応付けられます。

flowadm コマンドを使用してフローを識別する際に、帯域幅リソースの制御を課さないようにすることもできます。この場合、そのトラフィックタイプは独自のリソースやキューを取得するため、残りのネットワークトラフィックから隔離され、より監視しやすい決定性動作が実現されます。

フローの作成、変更、および削除は、大域ゾーン、非大域ゾーンのどちらでも行うことができます。ゾーン管理者は、自身の大域または非大域ゾーン内でしかフローを作成できません。ただし、次の段落で説明するように、大域ゾーンで作成されたフローを非大域ゾーンに移行することは可能です。管理者がフローの変更や削除を行えるのは、そのフローの作成元となる大域または非大域ゾーン内からだけです。大域ゾーンのユーザーは、大域ゾーン内とすべての非大域ゾーン内に存在している、システム上のすべてのフローを表示できます。非大域ゾーンのユーザーが表示できるのは、そのゾーン内のフローだけです。

管理者は、大域ゾーン内でのフローの作成が完了すると、そのフローに関連付けられたデータリンクを非大域ゾーンに割り当てることができます。そのような場合、関連するフローもその同じ非大域ゾーンに割り当てられます。この非大域ゾーンが停止すると、データリンクとその関連フローは大域ゾーンに戻されます。

同じ名前のフローでも、ゾーン名が異なれば区別されます。たとえば、`fastpak` という名前のフローを 3 つ作成する場合、`fastpak` をそれぞれ異なるゾーン内に格納すれば可能となります。たとえば、`zone1/fastpak`、`zone2/fastpak`、および `zone3/fastpak` はすべて有効なゾーン名です。

`flowadm` は、対応するオプションを備えた一連のサブコマンドとして実装されています。オプションについては各サブコマンドの文脈で説明します。

サブコマンド

サポートされているサブコマンドは次のとおりです。

`flowadm show-flow` [`-pP`] [`-o` *field*[,...]] [`-l` *link*] [*flow*]

すべてのフロー、1 つのリンクのすべてのフロー、指定されたフローのいずれかに対するフロー構成情報 (デフォルト) または統計情報を表示します。

`-o` *field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 `all` にする必要があります。見つかったフローごとに、次のフィールドを表示できます。

`flow`

フローの名前。

`link`

フローが存在するリンクの名前。

`ipaddr`

フローの IP アドレス。これは、フローの定義方法に応じて、ローカルかリモートのいずれかになります。

`transport`

使用されるプロトコルのレイヤーの名前。

`port`

フローのサービスのローカルポート。

`dsfield`

フローの差別化サービス値、および IP ヘッダーの差別化サービスフィールドのどのビットを対象とするかを指定するために `DSFIELD` 値とともに使用されるマスク。

`-p`、`--parseable`

安定したマシン解析可能な形式を使用して表示します。

`-P`、`--persistent`
永続的なフロープロパティーの情報を表示します。

`-l` *link*, `--link=`*link* | *flow*
指定されたリンク上のすべてのフローの情報または指定されたフローの情報を表示します。

`flowadm add-flow` [`-t`] [`-R` *root-dir*] `-l` *link* `-a` *attr=value*[,...] `-p` *prop=value*[,...] *flow*
システムにフローを追加します。フローはそのフロー属性とフロープロパティーによって識別されます。

特定のフローを識別するにあたり、その帯域幅リソースを制限できます。帯域幅制限を指定しない場合でも、そのトラフィックは一意のレイヤー 2、3、および 4 キューや処理用スレッドを取得するほか、NIC ハードウェアリソース (サポートされている場合) も取得します。このため、その選択されたトラフィックをほかのトラフィックから隔離することが可能となり、フローに対するほかのトラフィックからの影響を最小限に抑えることができます。

`-t`, `--temporary`
変更は一時的なものであり、リブート後は永続しません。永続がデフォルトです。

`-R` *root-dir*, `--root-dir=`*root-dir*
`flowadm` が永続的な作成を適用するべき代替ルートディレクトリを指定します。

`-l` *link*、`--link=`*link*
フローの追加先となるリンクを指定します。

`-a` *attr=value*[,...]、`--attr=`*value*
指定された値に設定される属性のコンマ区切りリスト。

`-p` *prop=value*[,...], `--prop=`*value*[,...]
指定された値に設定するプロパティーのコンマ区切りのリスト。

`flowadm remove-flow` [`-t`] [`-R` *root-dir*] `-l` {*link* | *flow*}
リンクまたは名前で特定された既存のフローを削除します。

`-t`, `--temporary`
変更は一時的なものであり、リブート後は永続しません。永続がデフォルトです。

`-R` *root-dir*, `--root-dir=`*root-dir*
`flowadm` が永続的な削除を適用するべき代替ルートディレクトリを指定します。

`-l` *link* | *flow*, `--link=`*link* | *flow*
リンクが指定されている場合、そのリンクからすべてのフローを削除します。単一のフローが指定されている場合は、そのフローだけを削除します。

`flowadm set-flowprop` [`-t`] [`-R` *root-dir*] `-p` *prop=value*[,...] *flow*

名前で指定されたフローで、1 つ以上のプロパティーの値を設定します。プロパティーの完全な一覧を取得するには、`show-flow` サブコマンドを使用します。

`-t`, `--temporary`

変更は一時的なものであり、リブート後は永続しません。永続がデフォルトです。

`-R` *root-dir*, `--root-dir=`*root-dir*

`flowadm` が永続的なプロパティー設定を適用するべき代替ルートディレクトリを指定します。

`-p` *prop=value*[,...], `--prop=`*value*[,...]

指定された値に設定するプロパティーのコンマ区切りのリスト。

`flowadm reset-flowprop` [`-t`] [`-R` *root-dir*] `-p` [*prop=value*[,...]] *flow*

指定されたフローで、1 つ以上のプロパティーをデフォルト値にリセットします。プロパティーが指定されていない場合、すべてのプロパティーがリセットされます。デフォルト値など、プロパティーの説明については、`show-flowprop` サブコマンドを参照してください。

`-t`, `--temporary`

リセットを一時的なものとして指定します。一時的なリセットは、次回のリブートまで保持されます。

`-R` *root-dir*, `--root-dir=`*root-dir*

`flowadm` が永続的なプロパティー設定を適用するべき代替ルートディレクトリを指定します。

`-p` *prop=value*[,...], `--prop=`*value*[,...]

リセットするプロパティーのコンマ区切りのリスト。

`flowadm show-flowprop` [`-cP`] [`-l` *link*] [`-p` *prop*[,...]] [*flow*]

すべてのフロー、指定されたリンクのフロー、指定されたフローのいずれかに対する 1 つ以上のプロパティーの現在値または永続的な値を表示します。

デフォルトでは、現在の値が表示されます。プロパティーが指定されていない場合、使用可能なすべてのフロープロパティーが表示されます。プロパティーごとに、次のフィールドが表示されます。

`FLOW`

フローの名前。

`PROPERTY`

プロパティーの名前。

`VALUE`

現在の (または永続的な) プロパティー値。値は、未設定の場合は `--` (ハイフン 2 つ) として表示され、未知の場合は `?` (疑問符) として表示されます。設定さ

れていないか、またはリセットされた永続的な値は -- と表示され、システムの DEFAULT 値 (もしあれば) を使用します。

DEFAULT
プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、-- (ハイフン 2 つ) が表示されます。

POSSIBLE
プロパティーに設定できる値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形として最小値と最大値が表示される可能性があります。指定可能な値が不明または無限の場合は、-- (ハイフン 2 つ) が表示されます。

フロープロパティーについては「フロープロパティー」の節で説明されています。

-c, --parseable
安定したマシン解析可能な形式を使用して表示します。

-P、--persistent
永続的なフロープロパティーの情報を表示します。

-p *prop*[,...], --prop=*prop*[,...]
表示するプロパティーのコンマ区切りのリスト。

フロー属性

flowadm コマンド内でフローを識別するフローオペランドは、次の一覧から選択した 1 つ以上のキーワード-値ペアのコンマ区切りのリストです。

local_ip[/*prefix_len*]
ローカル IP アドレスでネットワークフローを識別します。*value* はドット区切り 10 進表記の IPv4 アドレス、コロン区切り表記の IPv6 アドレスのいずれかでなければいけません。*prefix_len* は省略可能です。

prefix_len が指定された場合、それは、ifconfig(1M) や route(1M) のアドレスと同じ表記規則に従って、サブネットアドレスのネットマスクを記述します。これが指定されなかった場合、指定された IP アドレスはホストアドレスとみなされ、そのデフォルトプレフィックス長は、IPv4 アドレスでは /32、IPv6 では /128 になります。

remote_ip[/*prefix_len*]
リモート IP アドレスでネットワークフローを識別します。その構文は local_ip 属性と同じです

transport={tcp|udp|sctp|icmp|icmpv6}
使用するレイヤー 4 プロトコルを識別します。これは通常、特別な注意を払う必要のあるローカルまたはリモートのサービスを識別するために、local_port または remote_port と組み合わせて使用されます。

local_port
ローカルポートで指定されるサービスを識別します。

`remote_port`
リモートポートで指定されるサービスを識別します。

`dsfield`[:*dsfield_mask*]
RFC 2474 で定義されているとおりに 8 ビットの差別化サービスフィールドを識別します。

省略可能な *dsfield_mask* は、`dsfield` 値との比較時に差別化サービスフィールドのどのビットを対象とするかを指定するために使用されます。あるビット位置の値が `0` の場合、それは、そのビットの値を無視する必要があることを示し、`1` の場合はその逆を示します。このマスクの範囲は `0x01` から `0xff` です。*dsfield_mask* が指定されていなければ、デフォルトマスク `0xff` の値が使用されます。`dsfield` 値とマスクはどちらも 16 進数値でなければいけません。

サポートされている属性の組み合わせの種類は、次のとおりです。

```
local_ip[/prefixlen]=address
remote_ip[/prefixlen]=address
transport={tcp|udp|sctp|icmp|icmpv6}
transport={tcp|udp|sctp},local_port=port
transport={tcp|udp|sctp},remote_port=port
dsfield=val[:dsfield_mask]
```

ある特定のリンク上で、上の各組み合わせは相互に排他的です。複数の組み合わせを含むフローを作成しようとすると、失敗します。

制限

制限には、個別のフロー制限とゾーンごとのフロー制限があります。

個別のフロー制限

個々のフローに対する制限では、同じリンクに追加されたほかのフローを考慮する必要はありません。

各フローで属性を指定できるのは、1 回だけです。たとえば、次のコマンドは有効ではありません。

```
# flowadm add-flow -l vnic1 -a local_port=80,local_port=8080 httpflow
```

`transport` と `local_port` または `transport` と `remote_port`:

TCP、UDP、または SCTP フローの指定時には、ローカルポートまたはリモートポートを使用できます。ICMP または ICMPv6 フローではポートを指定できません。

`local_port`、`remote_port` のいずれかを指定する場合、トランスポートは TCP、UDP、または SCTP のいずれかでなければいけません。

次のコマンドは有効です。

```
# flowadm add-flow -l e1000g0 -a transport=udp udpflow
# flowadm add-flow -l e1000g0 -a transport=tcp,local_port=80 \
udp80flow
```

次のコマンドは有効ではありません。

```
# flowadm add-flow -l e1000g0 -a remote_port=25 flow25
# flowadm add-flow -l e1000g0 -a transport=icmpv6,remote_port=16 \
flow16
```

ゾーンごとのフロー制限

あるゾーン内で同じ名前のフローを 2 つ作成することはできません。リンクを指定してフローを追加した場合、そのフローを表示、変更、または削除する際にリンクを指定する必要はありません。

フロープロパティー

サポートされるフロープロパティーは次のとおりです。あるプロパティーを特定の値に設定できるかどうかは、ドライバやハードウェアによって異なることに注意してください。

`maxbw`
: フローの全二重帯域幅を設定します。この帯域幅は、いずれかのスケールサフィックス (Kbps、Mbps、または Gbps を示す `K`、`M`、または `G`) が付いた整数として指定されます。単位が指定されていない場合、入力値は Mbps として読み取られます。デフォルトでは、帯域幅の制限はありません。

使用例

例 1 ミッションクリティカルポートに関するポリシーの作成

次のコマンドは、HTTPS が専用の NIC ハードウェアとカーネル TCP/IP リソースを取得できるように、HTTPS サーバーのインバウンド HTTPS トラフィックに関するポリシーを作成しています。指定されている名前 `https-1` は、あとでポリシーの変更や削除を行う際に使用できます。

```
# flowadm add-flow -l bge0 -a transport=TCP,local_port=443 https-1
# flowadm show-flow -l bge0
FLOW         LINK         IP ADDR            PROTO  PORT  RPORT   DSFLD
https1       bge0         --                 tcp    443   --      --
```

例 2 既存のポリシーの変更による帯域幅リソース制御の追加

次のコマンドは、前述の例の `https-1` ポリシーを変更します。このコマンドは帯域幅制御を追加します。

```
# flowadm set-flowprop -p maxbw=500M https-1
# flowadm show-flow https-1
FLOW         LINK         IP ADDR            PROTO  PORT  RPORT   DSFLD
https1       bge0         --                 tcp    443   --      --

# flowadm show-flowprop https-1
FLOW        PROPERTY    VALUE     DEFAULT      POSSIBLE
https-1     maxbw       500       --           --
```

例 3　UDP 帯域幅使用量の制限

次のコマンドは、UDP プロトコルのポリシーを作成し、使用可能な帯域幅のうち、100 Mbps を超える量をそのプロトコルが消費できないようにしています。このフローには limit-udp-1 という名前が付けられています。

```
# flowadm add-flow -l bge0 -a transport=UDP -p maxbw=100M, \
limit-udp-1
```

例 4　dsfield 属性を使用したポリシーの設定

次のコマンドは、帯域幅 500Mbps の EF PHB (RFC 2598 の DSCP 値 101110) に対するポリシーを設定します。このフローでは、dsfield の値が 0x2e (101110)、dsfield_mask が 0xfc になっています (下位 2 ビットを無視する必要があるため)。

```
# flowadm add-flow -l bge0 -a dsfield=0x2e:0xfc \
-p maxbw=500M efphb-flow
```

例 5　複数ゾーン内のフローの表示

次のコマンドは、同じ名前を持つ 2 つのフローを表示しています。最初のフローは大域ゾーン内に存在し、2 番目のフローはゾーン zone1 内に存在しています。このコマンドは大域ゾーンから呼び出されているため、ユーザーはシステム上のすべてのフローを表示できます

```
# flowadm show-flow
FLOW        LINK        IPADDR            PROTO  LPORT   RPORT   DSFLD
tcpflow     e1000g2     --                tcp    --      --      --
zone1/tcpflow e1000g1   --                tcp    --      --      --
```

終了ステータス

0
　アクションはすべて正常に実行されました。

>0
　エラーが発生しました。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/network
インタフェースの安定性	確実

関連項目

acctadm(1M), dladm(1M), flowstat(1M), ifconfig(1M), prstat(1M), route(1M), attributes(5), ifconfig(1M)

注意事項

以前のリリースの flowadm に存在していた show-usage サブコマンドは、flowstat(1M) -h コマンドで置き換えられました。

名前

flowstat – フロー統計の報告

形式

```
flowstat [-r | -t] [-i interval] [-l link] [flow]

flowstat [-S] [-A] [-i interval] [-p] [ -o field[,...]]
     [-u R|K|M|G|T|P] [link] [-l link] [flow]

flowstat -h [-a] -f filename [-d] [-F format] [-s time]
     [-e time] [flow]
```

機能説明

flowstat コマンドは、ユーザー定義フローに関する実行時統計情報を報告します。flowadm show-flow は、このコマンド向けにフロー名の情報を提供します。

オプション

flowstat コマンドには次のオプションとオペランドがあります。これらは、後述の「サブコマンド」に示されているいくつかのコマンド形式の間で共通です。

サブコマンド

flowstat は、次のコマンド形式をサポートします。

flowstat [-r | -t] [-i *interval*] [-l *link*] [*flow*]

この形式のコマンドは、すべてのフローを繰り返し検査して統計情報を報告します。出力は、フロー使用率の高い順にソートされます。フローが指定されていない場合、システムはすべてのフローの統計情報を表示します。

-r
受信側の統計情報のみを表示します。受信されたバイト数とパケット数や、ドロップなどが含まれます。完全な一覧については、例を参照してください。

-t
送信側の統計情報のみを表示します。送信されたバイト数とパケット数や、ドロップなどが含まれます。後述の例を参照してください。

-i *interval*
統計情報が更新される間隔を秒単位で指定します。デフォルトの間隔は 1 秒です。

-l *link* | *flow*]
指定されたリンク上のすべてのフローの統計情報または指定されたフローの統計情報を表示します。

flowstat [-S] [-A] [-i *interval*] [-p] [-o *field*[,...]] [-u R|K|M |G|T|P] [-l *link*] [*flow*]

この形式のコマンドでは、どの統計情報を表示するかを指定できます。

-A
このフローのすべての統計情報フィールドをダンプします。このコマンドの出力統計情報には、ほかのすべての flowstat コマンドによって報告されるすべての統計情報が含まれます。

-i *interval*
統計情報が更新される間隔を秒単位で指定します。デフォルトの間隔は 1 秒です。

{-l *link*} | *flow*
指定されたリンク上のすべてのフローまたは指定されたフローの統計情報を表示します。

-o *field*[,...]
表示する出力フィールドのコンマ区切りリストを表示します (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはサポートされるすべてのフィールドを表示するための特殊な値 all にする必要があります。

サポートされる RX フィールドの一覧:

- flow
- rbytes
- ipkts
- idrops

TX フィールドの一覧:

- flow
- obytes
- opkts
- odrops

-p
安定したマシン解析可能な形式を使用して出力します。

-S
prstat(1M) が CPU 使用率をプロセス別に表示するのと似た方法で、フロー別のネットワーク使用率を継続的に表示します。

-u R|K|M|G|T|P
使用した場合、すべての統計情報の表示単位を選択できます (たとえば、R: 生の数値、K: キロビット、M: メガビット、T: テラビット、P: ペタビット)。使用しなかった場合、必要に応じて異なる単位を使用して統計情報が表示されます。

flowstat -h [-a] -f *filename* [-d] [-F *format*] [-s *time*] [-e *time*] [*flow*]
格納されている拡張アカウンティングファイルからネットワーク使用率の履歴を表示します。この構文を使用するには、acctadm(1M) によってネットワークアカウンティングがあらかじめ構成され、有効になっている必要があります。デフォルトでは、拡張アカウンティングが有効になっていた全期間における既存のリンクのネットワーク使用率の要約が出力されます。

-a
拡張アカウンティングが有効になっている場合に、特定の期間におけるすべてのネットワーク使用率の履歴を表示します。これには、すでに削除されたフローの使用率の情報が含まれます。

-f *filename*
ネットワークフロー使用率の履歴の拡張アカウンティングレコードが読み取られるファイルを指定します。

-d
ロギング情報が存在する日付を表示します。日付は、*mm*/*dd*/ *yyyy* の形式で表示されます。

-F *format*
ネットワークフロー使用率の履歴情報の出力形式を指定します。`gnuplot` が唯一のサポートされる形式です。

-s *time*
-e *time*
データ表示の開始時刻と停止時刻を指定します。時刻は、*MM*/*DD*/ YYYY,hh:mm *ss* の形式で指定します。*hh* は 24 時間表記を使用します。

オペランド

`flowstat` コマンドの各形式にはオペランドが 1 つ含まれていますが、これは省略可能です。

flow
指定された場合、指定されたフローについてのみ報告します。それ以外の場合は、すべてのフローについて報告します。フローは *zonename*/*flowname* という形式の名前を持ちます。*zonename* 修飾子の付いていない *flowname* は、大域ゾーン内に存在するものと理解されます。

使用例

例 1　統計情報の表示

すべてのフローの統計情報を表示するには、次のコマンドを入力します。統計情報は、3 桁の数値に該当する単位が付けられた形で表示されます。デフォルトの間隔は 1 秒です。

```
# flowstat -i 1
 FLOW   IPKTS  RBYTES  IDROPS   OPKTS  OBYTES  ODROPS
flow1 528.54K 787.39M       0 179.39K  11.85M       0
flow2 742.81K   1.10G       0       0       0       0
flow3       0       0       0       0       0       0
flow1  67.73K 101.02M       0  21.04K   1.39M       0
flow2       0       0       0       0       0       0
flow3       0       0       0       0       0       0
            .       .       .       .       .       .
            .       .       .       .       .       .
            .       .       .       .       .       .
```

例 2　受信側の統計情報の表示

次のコマンドは受信側の統計情報を表示します

```
# flowstat -r
 FLOW   IPKTS  RBYTES  IDROPS
```

例 2　受信側の統計情報の表示　　*(続き)*

```
flow1   4.01M   5.98G       0
flow2 742.81K   1.10G       0
flow3       0       0       0
```

例 3　送信側の統計情報の表示

次のコマンドは、送信側の統計情報を 5 秒間隔で表示します。

```
# flowstat -t
 FLOW   OPKTS  OBYTES  ODROPS
flow1  24.37M   1.61G       0
flow2       0       0       0
flow3       4     216       0
```

例 4　統計情報の特定のセットの表示

次のコマンドは、指定された一連の統計情報フィールドを表示します。

```
# flowstat -o FLOW,IPKTS
 FLOW   IPKTS
flow1  68.58M
flow2 742.81K
flow3       4
```

例 5　ネットワーク使用率の履歴の表示

フロー使用率の統計は、拡張アカウンティング機能 acctadm(1M) を使用すると格納できます。

```
# acctadm -e extended -f /var/log/net.log net
# acctadm net
         Network accounting: active
    Network accounting file: /var/log/net.log
  Tracked Network resources: extended
Untracked Network resources: none
```

保存されたデータを取得するには、次のようにします。

```
# flowstat -h -f /var/log/net.log
LINK      DURATION  IPACKETS RBYTES      OPACKETS OBYTES      BANDWIDTH
flowtcp   100       1031     546908      0        0            43.76Kbps
flowudp   0         0        0           0        0             0.00Mbps
```

2008 年 2 月 19 日 10 時 38 分 46 秒から同じ日の 10 時 40 分 06 秒までの flowtcp のロギング情報を表示するには、次のようにします。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp
FLOW      START       END         RBYTES   OBYTES     BANDWIDTH
flowtcp   10:39:06    10:39:26    1546     6539        3.23 Kbps
```

例 5　ネットワーク使用率の履歴の表示　　(続き)

```
flowtcp   10:39:26    10:39:46    3586     9922        5.40 Kbps
flowtcp   10:39:46    10:40:06    240      216       182.40 bps
flowtcp   10:40:06    10:40:26    0        0           0.00 bps
```

上と同じ出力情報をプロットファイルとして生成するには、次のようにします。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```

終了ステータス

0
　アクションはすべて正常に実行されました。

>0
　エラーが発生しました。

属性

次の属性についての詳細は、attributes(5) を参照してください。

/usr/sbin

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照

画面出力は「不確実」です。呼び出しは「確実」です。

関連項目

acctadm(1M), dladm(1M), dlstat(1M), flowadm(1M), ifconfig(1M), prstat(1M), route(1M), attributes(5), dlpi(7P)

名前

fmadm – 障害管理構成ツール

形式

`fmadm [-q] [subcommand [arguments]]`

機能説明

fmadm ユーティリティーは、管理者とサービス担当者が Solaris Fault Manager (fmd(1M)) によって管理されているシステム構成パラメータを表示および変更するために使用できます。 fmd は、システムソフトウェアが検出した問題に関する遠隔測定情報を受け取り、問題を診断し、障害のあるコンポーネントの無効化などの予測的自己修復操作を開始します。

fmadm を使って次の操作を実行できます。

- 障害管理に現在関与している一連の診断エンジンおよびエージェントの表示
- 障害があると診断されたシステムコンポーネントのリストの表示
- 前述のエンティティーに関連する管理タスクの実行

Fault Manager ができるだけ多くの操作を自動化しようとするため、通常は fmadm を使用する必要はありません。Fault Manager が管理者、サービス修復技術者、または Oracle の助けを必要とする場合は、その必要性を示すメッセージが生成されます。また、Sun の Web サイトのナレッジ記事が参照先として示されます。この Web サイトでは、fmadm またはほかの障害管理ユーティリティーを使った詳細情報の収集や追加タスクの実行を求められることがあります。fmd(1M)、fmdump(1M)、および fmstat(1M) の各ドキュメントには、障害管理操作を監視するツールに関する詳しい説明があります。

Fault Manager が行う処理の 1 つに、コンポーネントの位置の追跡があります。シャーシレベルでは、fmadm *-alias サブコマンドはシャーシの *product-id.chassis-id* から *alias-id* へのマッピングを管理します。管理された *alias-id* は、なんらかの意味のある方法でシャーシの物理的な位置を記述するためのものです。

fmadm ユーティリティーを使用するユーザーは、SYS_CONFIG 特権を持っている必要があります。Solaris の特権を構成する方法の詳細については、『Oracle Solaris の管理: セキュリティーサービス』 を参照してください。fmadm load サブコマンドを使用するユーザーは、すべての特権を持っている必要があります。

サブコマンド

fmadm で使用できるサブコマンドは、次のとおりです。一部のサブコマンドは追加のオプションやオペランドを受け入れるか、または追加のオプションやオペランドが必要です。load、unload、reset、および rotate は、トレーニングを受けた技術担当者向けのサブコマンドです。これらのサブコマンドを使用する際には、ナレッジベース記事などの具体的な指針に従うことをお勧めします。

fmadm acquit *fmri* | *label* [*uuid*]

Fault Manager に、指定したリソースが *uuid* で識別される障害イベントで、または UUID が指定されていない場合は、検出されたどの障害でも疑わしいとは見なされないことを通知します。fmadm acquit サブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。管理者

は、以前に障害が発生したリソースを再度有効にするために、追加のコマンドを適用しなければならない場合があります。

`fmadm acquit` *uuid*
Fault Manager に、*uuid* で識別された障害イベントを安全に無視できることを通知します。`fmadm acquit` サブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。管理者は、以前に障害が発生したリソースを再度有効にするために、追加のコマンドを適用しなければならない場合があります。

`fmadm config`
Fault Manager 自体の構成を表示します。これには、各コンポーネントモジュールのモジュール名、バージョン、および説明が含まれます。Fault Manager のモジュールは、システムに存在するハードウェアおよびソフトウェアに対して自動診断、自己修復、メッセージ配信などのサービスを提供します。

`fmadm faulty` [`-afgiprsv`] [`-n` *max*] [`-u` *uid*]
Fault Manager で現在障害があると見なされているリソースのステータス情報を表示します。

次のオプションがサポートされています。

`-a`
すべての障害を表示します。 `fmadm faulty` コマンドのデフォルトでは、現時点で存在し、障害のあるリソースに関する出力が一覧表示されます。`-a` オプションを指定すると、自動的に修正された障害や、復旧操作を必要としない障害も含め、Fault Manager によってキャッシュされたすべてのリソース情報が表示されます。この一覧表示には、システムにすでに存在しない可能性があるリソースの情報も含まれます。

`-f`
障害のある `FRU` (現場交換可能ユニット) を表示します。

`-g`
FRU、クラス、および障害メッセージが同じ障害をグループ化します。

`-i`
Fault Manager 内の各リソースの永続的キャッシュ識別子を表示します。

`-n` *max*
障害またはリソースが `-a` または `-g` オプションでグループ化されている場合は、出力を *max* 個のエントリに制限します。

`-p`
各障害の間にフォームフィードを挿入して出力をページャーを介してパイプします。

`-r`
障害管理リソースを識別子 (FMRI) と障害管理状態とともに表示します。

`-s`
障害イベントごとに 1 行の障害要約を表示します。

`-u` *uid*
指定された uid の障害のみを表示します。

`-v` 完全な出力を表示します。

障害の原因が、別のクラスまたは別の FRU のどちらかに複数存在する場合は、確実性の割合 (パーセント) が表示されます。同じ FRU 上に複数のリソースがあり、障害がその FRU に関連していることが 100% 確実でない場合は、原因がその FRU にある確実性の最大の割合 (パーセント) が表示されます。

Fault Manager は、受け取った遠隔測定情報を持つすべてのリソースに次の状態を割り当てます。

`ok`
リソースが存在し、使用中であり、Fault Manager に関するかぎり既知の問題はありません。

`unknown`
リソースが存在しないか、または使用不能ですが、既知の問題はありません。これは、管理者によってリソースが無効化または構成解除されたことを示している場合があります。詳細については、適切な管理ツールを参照してください。

`faulted`
リソースは存在しますが、Fault Manager の診断によって 1 つ以上の問題が見つかったため、使用不能です。システムにこれ以上の損害を与えないように、このリソースは無効化されています。

`degraded`
リソースが存在し、使用できますが、Fault Manager の診断によってこのリソースに 1 つ以上の問題が見つかりました。

影響を受けるすべてのリソースが同じ状態にある場合は、これがリストの最後にあるメッセージに反映されます。それ以外の場合は、影響を受ける各リソースのあとに状態が表示されます。

`fmadm flush` *fmri*
Fault Manager によってキャッシュされたリソースの情報を FMRI で指定してフラッシュします。このサブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。Fault Manager がキャッシュを自動的に最新の状態に維持するため、通常はこのコマンドを使用する必要はありません。キャッシュから障害のあるリソースをフラッシュした場合、管理者は指定したリソースを有効にするために追加のコマンドを適用しなければならないことがあります。

`fmadm load` *path*
指定された Fault Manager モジュールを読み込みます。*path* は、絶対パスで指定し、モジュール用に定義されたいずれかのディレクトリに存在するモジュールを参照する必要があります。Fault Manager が Solaris の初期起動時や必要な場合に自動的にモジュールを読み込むため、通常はこのコマンドを使用する必要はありません。

`fmadm unload` *module*
指定された Fault Manager モジュールを読み込み解除します。*module* は、`fmadm config` の出力に表示されたベース名を使って指定します。Fault Manager がシステム構成に基づいて自動的にモジュールの読み込みと読み込み解除を行うため、通常はこのコマンドを使用する必要はありません。

`fmadm repaired` *fmri* | *label*
Fault Manager に、指定したリソースに対して修復手順が実行されたことを通知します。`fmadm repaired` サブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。管理者は、以前に障害が発生したリソースを再度有効にするために、追加のコマンドを適用しなければならない場合があります。

`fmadm replaced` *fmri* | *label*
Fault Manager に、指定したリソースが交換されたことを通知します。このコマンドは、Fault Manager が交換を自動的に検出できない場合に使用してください。`fmadm replaced` サブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。管理者は、以前に障害が発生したリソースを再度有効にするために、追加のコマンドを適用しなければならない場合があります。

`fmadm reset` [`-s` *serd*] *module*
指定された Fault Manager モジュールまたはモジュールのサブコンポーネントをリセットします。`-s` オプションを指定すると、モジュール内部の指定した Soft Error Rate Discrimination (SERD) エンジンがリセットされます。`-s` オプションを指定しないと、モジュール全体がリセットされ、モジュールに関連付けられたすべての永続状態が削除されます。`fmadm reset` サブコマンドは、Sun のマニュアルに記載されている修復手順で指示された場合にのみ使用してください。Fault Manager がモジュールを自動的に管理するため、通常はこのコマンドを使用する必要はありません。

`fmadm rotate errlog | fltlog | infolog | infolog_hival`
`rotate` サブコマンドは、`logadm` が有効なログファイルを正しくローテーションできるようにするための、`logadm(1M)` のヘルパーコマンドです。これは、直接呼び出すようには設計されていません (直接呼び出すとログ履歴が失われる可能性が高くなります)。現在のログファイルのサイズが 0 でない場合に、適切なログファイルがローテーションされるようにするには、次のいずれかのコマンドを使用します。

```
# logadm -p now -s 1b /var/fm/fmd/errlog
# logadm -p now -s 1b /var/fm/fmd/fltlog
# logadm -p now -s 1b /var/fm/fmd/infolog
# logadm -p now -s 1b /var/fm/fmd/infolog_hival
```

`fmadm add-alias` *product-id.chassis-id alias-id* ['*comment*']
`add-alias` サブコマンドは、*alias-id* を *product-id.chassis-id* シャーシの管理された別名として確立するために使用します。管理された別名が定義されている場

合、シャーシの /dev/chassis devchassis(7FS) 名前空間表現では、*product-id.chassis-id* ではなく、より意味のある *alias-id* が使用されます。

```
# fmadm add-alias SUN-Storage-J4410.1039QAQ007 RACK29.U25-28
```

上記のコマンドでは、新しいマッピングが既存のマッピングと競合しないことを検証します。競合する場合は、マッピングの変更は行われません。このサブコマンドは、関連付けられた名前空間の更新が完了したときに完了します。更新された名前空間が新しい *alias-id* を使用しない場合、警告が出力されますが、マッピングは更新されます。名前空間の更新に時間がかかりすぎる場合、警告が出力されます。

省略可能な注釈を指定した場合、注釈は保持され、後続の lookup-alias または list-alias コマンドによって表示されます。

fmadm remove-alias *alias-id* | *product-id.chassis-id*

remove-alias サブコマンドは、*product-id.chassis-id* から *alias-id* へのマッピングを削除するために使用します。

```
# fmadm remove-alias RACK29.U25-28
```

上記のサブコマンドは、関連付けられた名前空間の更新が完了したときに完了します。

fmadm lookup-alias *alias-id* | *product-id.chassis-id*

lookup-alias サブコマンドは、現在のマッピングの内容を確認するために使用できます。コマンドの例を次に示します。

```
# fmadm lookup-alias SUN-Storage-J4410.1039QAQ007
```

fmadm list-alias

list-alias サブコマンドは、すべての注釈とマッピングを表示するために使用します。

fmadm sync-alias

sync-alias サブコマンドは、一連のマッピングを一括で手動インポートするために使用します。現在のマッピングは 2 つのコピーが保持されます。

- /etc/dev/chassis_aliases
- /etc/dev/.chassis_aliase

一連のマッピングを一括でインポートするには、/etc/dev/chassis_aliases ファイルを更新してから、sync-alias を実行します。

オプション

次のオプションがサポートされています。

-q　非出力モードを設定します。操作が成功したときに、fmadm は標準出力に結果メッセージを出力しません。

オペランド

次のオペランドがサポートされています。

cmd 「サブコマンド」に示されているサブコマンドの名前。

args 選択したサブコマンドに対応する 1 つ以上のオプションまたは引数 (「サブコマンド」に示されているもの)。これらの引数としては、fmri、uuid、label などが挙げられます。これらは、fmadm サブコマンドの対象となるリソースを識別します。ある対象リソースの fmri、uuid、および label を取得するには、fmadm faulty を使用します。「使用例」を参照してください。一般に、これらのオペランドの中でもっともユーザーにとってわかりやすいのは、label です。

使用例

例 1 faulty サブコマンドの呼び出し

次のコマンドは faulty サブコマンドを呼び出すことで、あるコンポーネントの uuid、label、および fmri を表示しています。

```
# fmadm faulty
------------ ------------------------------------  ------------ ---------
TIME         EVENT-ID                              MSG-ID       SEVERITY
------------ ------------------------------------  ------------ ---------
Sep 09 16:15 96609fae-113c-e48c-b1cf-ebf4b0902d72  DISK-8000-3E Critical
                                                                injected

Host      : x4170-brm-02
Platform  : SUN-FIRE-X4170-SERVER     Chassis_id  : 0920XF508B
Product_sn:

Fault class: fault.io.scsi.cmd.disk.dev.rqs.derr
Affects : dev:///:devid=id1,sd@n5000c5000940edbb//scsi_vhci/disk@g5000c\
5000940edbb  out of service, but associated components no longer faulty
FRU     : "DISK 11"
(hc://:product-id=SUN-Storage-J4410:server-id=:chassis-id=:serial=000930\
G01CN4----3SJ01CN4:part=SEAGATE-ST330057SSUN300G:revision=0205/\
ses-enclosure=0/bay=11/disk=0)
replaced
...
...
```

前述の出力で、uuid は EVENT-ID 列の最初の項目 96609fae-113c-e48c-b1cf-ebf4b0902d72 です。label は、FRU 行の DISK 11 です。fmri は次のとおりです。

```
dev:///:devid=id1,sd@n5000c5000940edbb//scsi_vhci/disk@g5000c\
5000940edbb

hc://:product-id=SUN-Storage-J4410:server-id=:chassis-id=:serial=000930\
G01CN4--------3SJ01CN4:part=SEAGATE-ST330057SSUN300G:revision=0205/\
ses-enclosure=0/bay=11/disk=0)
```

例 1 faulty サブコマンドの呼び出し (続き)

fmdump -v でも次のように、同じ値を表示できます。

```
# fmdump -v
Sep 09 16:15:36.9252 96609fae-113c-e48c-b1cf-ebf4b0902d72 DISK-8000-3E \
Diagnosed 100%  fault.io.scsi.cmd.disk.dev.rqs.derr

Problem in: hc://:scheme=:product-id=SUN-Storage-J4410:chassis-id=:\
server-id=/ses-enclosure=0/bay=11/disk=0

Affects: dev:///:devid=id1,sd@n5000c5000940edbb//\
scsi_vhci/disk@g5000c5000940edbb
FRU: hc://:product-id=SUN-Storage-J4410:server-id=:chassis-id=:\
serial=000930G01CN4--------3SJ01CN4:part=SEAGATE-ST330057SSUN300G:\
revision=0205/ses-enclosure=0/bay=11/disk=0

Location: DISK 11
```

label がもっとも使用しやすい識別子です。

例 2 モジュール名の取得

次のコマンドは、各コンポーネントのモジュール名を表示しています。モジュール名は、fmadm unload コマンドの入力として指定します。

```
# fmadm config
MODULE                   VERSION STATUS  DESCRIPTION
cpumem-retire            1.1     active  CPU/Memory Retire Agent
disk-transport           1.0     active  Disk Transport Agent
eft                      1.16    active  eft diagnosis engine
..
```

終了ステータス

次の終了値が返されます。

0 正常終了。

1 エラーが発生しました。エラーには、fmd との通信の失敗や、要求された操作の実行に必要な特権の不足などが含まれます。

2 無効なコマンド行オプションが指定されました。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management
インタフェースの安定性	下記を参照

コマンド行オプションは「確実」です。人間が読める形式の出力は非インタフェースです。

関連項目

fmd(1M), fmdump(1M), fmstat(1M), logadm(1M), syslogd(1M), attributes(5), devchassis(7FS)

『Oracle Solaris の管理: 一般的なタスク』

名前

fmd – Fault Manager デーモン

形式

`/usr/lib/fm/fmd/fmd [-V] [-f file] [-o opt=val] [-R dir]`

機能説明

fmd は、各 Solaris システムのバックグラウンドで実行されるデーモンです。 fmd は、システムソフトウェアが検出した問題に関する遠隔測定情報を受け取り、問題を診断し、障害のあるコンポーネントの無効化などの予測的自己修復操作を開始します。Fault Manager は必要に応じて、問題を検出したことを管理者に通知するために、syslogd(1M) サービスへのメッセージ送信も行います。このメッセージは、Oracle の Web サイト (https://support.oracle.com) のナレッジ記事へのリンクを管理者に提供します。この記事は、問題の影響や適切な対処方法について詳しく説明しています。

Fault Manager によって診断された各問題には、汎用一意識別子 (UUID) が割り当てられます。UUID は、任意の一連のシステムの中でこの特定の問題を一意に識別します。fmdump(1M) ユーティリティーを使用すると、Fault Manager によって診断された問題やその UUID、ナレッジ記事のメッセージ識別子を一覧表示できます。fmadm(1M) ユーティリティーを使用すると、システム上で障害があると見なされているリソースを表示できます。fmstat(1M) ユーティリティーを使用すると、Fault Manager によって保持されている統計情報を報告できます。Fault Manager は Solaris の起動時に自動的に開始されるため、fmd コマンドを直接使用する必要はありません。Sun の Web サイトでは、Solaris の Fault Manager でどのような機能が現在使用可能になっているかについて、詳しく説明しています。

通知サービス

syslog (パッケージ system/fault-management)

新しい診断の標準の通知メカニズムは syslog 経由での通知ですが、その際、fmd 自体と同じパッケージに同梱されている syslog-msgs fmd モジュールが使用されます。

デフォルトでは、次の表に示すような syslog の機能や重要度を使用したメッセージが syslog-msgs 経由で送信されるのは、新しい問題診断に対してだけです。管理者は svccfg(1M) を使用することで、問題解決のライフサイクルに含まれるその他のイベントについても syslog-msgs 経由でメッセージが送信されるように要求できます。

```
# svccfg setnotify event syslog:{active|inactive}
```

詳細は、svccfg(1M) を参照してください。

Event	Disposition	Facility	Severity
problem-diagnosed	active	LOG_DAEMON	LOG_ERR
problem-updated	inactive	LOG_DAEMON	LOG_NOTICE
problem-repaired	inactive	LOG_DAEMON	LOG_NOTICE
problem-resolved	inactive	LOG_DAEMON	LOG_NOTICE

電子メール (パッケージ system/fault-management/smtp-notify)

電子メール経由での通知もオプションの 1 つですが、その場合は追加のパッケージをインストールする必要があります。SMF サービス

svc:/system/fm/smtp-notify:default はパッケージ system/fault-management/smtp-notify によって配布され、通知設定は svccfg(1M) によって構成されます。詳細は、smtp-notify(1M) を参照してください。svccfg(1M) を使用すると、前述の問題ライフサイクルイベント (problem-diagnosed など) の通知設定を構成できるだけでなく、SMF インスタンスの状態遷移やその他のイベントを通知するようにこのメカニズムを構成することもできます。

SNMP (パッケージ system/fault-management/snmp-notify)
SNMP トラップを使用した新しいイベントの通知は、パッケージ system/fault-management/snmp-notify によって提供されるオプションです。サービス svc:/system/fm/snmp-notify:default は、問題ライフサイクルの SNMP トラップや、その他の指定されたイベント (SMF インスタンス状態遷移イベントの通知が構成されている場合はそうしたイベントなど) を発行する役割を果たします。詳細は、snmp-notify(1M) を参照してください。

Solaris の大域ゾーンと非大域ゾーン

Fault Manager サービス svc:/system/fmd:default サービスは、Solaris の大域ゾーンと非大域ゾーンの両方で構成されます。非大域ゾーンではハードウェア指向のさまざまな Fault Manager モジュールは提供されないため、そこで実行されるのは、機能を縮小した Fault Manager です。非大域ゾーンでは、Fault Manager はソフトウェアイベントに焦点を絞ります。

オプション

以下のオプションがサポートされています。

-f *file*
Fault Manager の任意のデフォルト構成ファイルを検索する前に、指定された構成ファイル *file* を読み取ります。

-o *opt=value*
指定された Fault Manager オプションを指定された値に設定します。Fault Manager オプションは現在「非公開」インタフェースになっています。「非公開」インタフェースについては、attributes(5) を参照してください。

-R *dir*
Fault Manager によって評価されるすべてのパス名で、デフォルトのルート (/) の代わりに指定されたルートディレクトリを使用します。

-V
Fault Manager のバージョンを標準出力に出力して終了します。

終了ステータス

次の終了値が返されます。

0 正常終了

1 遠隔測定用トランスポートのオープン失敗など、何らかのエラーが発生したために、Fault Manager の初期化が失敗しました。

2 無効なコマンド行オプションが指定されました。

ファイル

/etc/fm/fmd	Fault Manager の構成ディレクトリ
/usr/lib/fm/fmd	Fault Manager のライブラリディレクトリ
/var/fm/fmd	Fault Manager のログディレクトリ

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management
インタフェースの安定性	確実

関連項目

svcs(1), fmadm(1M), fmdump(1M), fmstat(1M), smtp-notify(1M), snmp-notify(1M), svccfg(1M), syslogd(1M), attributes(5), smf(5)

http://www.sun.com/msg/

注意事項

Fault Manager はサービス管理機能 smf(5) により、次のサービス識別子の下で管理されます。

```
svc:/system/fmd:default
```

サービスのステータスを照会するには、svcs(1) コマンドを使用します。管理者は、Fault Manager サービスを無効にするべきではありません。

名前

fmdump – 障害管理ログビューア

形式

```
fmdump [[-e | -i | -I] | -A ] [-f] [-mvVp] [-c class] [-R root]
     [-t time] [-T time] [-u uid] [-n name[.name]*[=value]]
     [-E ENA] [file] ...
```

機能説明

fmdump ユーティリティーは、Solaris Fault Manager (fmd(1M)) に関連付けられたログファイルの内容を表示するために使用できます。Fault Manager は各 Solaris システムのバックグラウンドで実行されます。Fault Manager は、システムソフトウェアが検出した問題に関する遠隔測定情報を受け取り、問題を診断し、障害のあるコンポーネントの無効化などの予測的自己修復操作を開始します。

fmdump ユーティリティーは、Fault Manager の主管理インタフェースとして設計されたものではありません。その目的には、fmadm を使用してください (fmadm(1M) を参照)。fmdump ユーティリティーは、新たな解釈をほとんど行うことなく Fault Manager の履歴ログを単純にダンプするほか、何の説明もなしに実装の詳細情報を含める可能性があります。後述の「属性」の節を参照してください。

Fault Manager は次のように、複数のセットのログファイルを維持しますが、これらは主にサービス担当者向けのものです (場合によっては管理者も使用可)。

エラーログ　エラー遠隔測定 (システムによって検出された問題の兆候) を記録するログ。

情報ログ　情報イベントを記録するログ。これは、高価値情報イベントとその他の情報イベントという 2 セットのログファイルとして実現されています。

障害ログ　障害診断情報 (エラーログや情報ログに記録された兆候を説明すると考えられる問題) を記録するログ。

1 つのログファイルセットは、現在のアクティブログファイルと、そのセット内のおそらくいくつかのローテーションされた古いログファイルとから構成されます。すべてのログは logadm(1M) で管理され、/etc/logadm.conf 内にエントリを持ちます。

fmdump ユーティリティーは、対象セットの現在のログファイルとローテーションされたすべてのログファイルをダンプします。したがって、ログ履歴の全体が表示されることになります。特に障害ログの場合、fmdump が、進行中の問題だけにとどまらず、かつて診断されたことのあるすべての問題を表示する点を認識することが重要です。(その情報を取得するには fmadm faulty を使用します。)

デフォルトでは、fmdump は障害ログの内容を表示します。障害ログには、Fault Manager またはそのコンポーネントモジュールによって行われた個々の診断の結果が記録されます。エラーログを選択するには -e を使用し、情報ログの場合は -i を、高価値情報ログの場合は -I を、それぞれ使用します。あるいは、コマンド行で特定のログファイルパスを [*file*] として指定することもできます (この場合、そのファイルだけがダンプされ、ローテーション版のログは検索されません)。さら

に、オプション -A を使用することで、すべてのログを集約したり、コマンド行に設定されたログファイルパスのセットを集約したりすることもできます。

fmdump のデフォルト表示の例を次に示します。

```
# fmdump
TIME                 UUID                                 SUNW-MSG-ID EVENT
Mar 23 14:06:35.2682 0a11a1a7-a8ce-c941-8527-8d7a9d320071 ZFS-8000-CS Diagnosed
Mar 25 14:51:41.2261 0a11a1a7-a8ce-c941-8527-8d7a9d320071 FMD-8000-4M Repaired
Mar 25 14:51:41.2523 0a11a1a7-a8ce-c941-8527-8d7a9d320071 FMD-8000-6U Resolved
May 31 23:35:39.9146 c63ac52e-506b-c1cc-e965-ff3b8544490d SMF-8000-YX Diagnosed
...
```

(80 文字以下の画面では出力が折り返されます。)

この場合は障害ログがダンプされます。ほかのログを選択するオプションや引数がコマンド行に指定されていないためです。障害ログには、Fault Manager やそのコンポーネントモジュールによって診断された問題のライフサイクル (初期の問題診断から、問題解決まで) が記録されます。

障害ログに記録された個々の問題は、次の要素で識別されます。

- イベントのタイムスタンプ。問題のライフサイクルの状態変化を記述します。
- 汎用一意識別子 (UUID)。一連のシステム間でその問題を一意に識別するために使用できます。ある特定の問題の問題ライフサイクル状態変化を記述するイベントはすべて同じ UUID を使用します (前述の例でも、初期診断やそのあとの修復や解決がすべて同じ問題 UUID を参照していることがわかります)。
- メッセージ識別子。Sun の Web サイト (http://www.oracle.com/us/sun/msg/) にある対応するナレッジ記事にアクセスするために使用できます。

問題に対して管理者やサービス担当者による処置が必要な場合や、問題がシステムの動作に影響する場合は、人間が読める形式のメッセージが Fault Manager から syslogd(1M) に出力されます。このメッセージには、問題の要約と Sun Web サイト (http://www.oracle.com/us/sun/msg/) のナレッジ記事への参照が含まれています。fmdump ユーティリティーで syslog への書き込みと似た形式で fltlog エントリをダンプするには、-m オプションを使用します。

-v および -V オプションを使って、1 行だけの要約から、ログに記録された各イベントの詳細情報のレベルまで、表示を展開できます。-p オプションを -V と組み合わせて使用すると、「より高品質の」出力を要求できます。

-c、-t、-T、-n、および -u オプションを使って、指定した class、時間範囲、または uuid と一致するイベントだけを選択することにより、出力をフィルタ処理できます。コマンド行に複数のフィルタオプションを指定すると、各オプションの論理 AND によって選択されたイベントだけを表示するようにオプションが組み合わされ

ます。コマンド行に同じフィルタオプションを複数回指定すると、各オプションの論理 OR によって選択されたイベントを表示するように同類のオプションが組み合わされます。次に例を示します。

```
# fmdump -u uuid1 -u uuid2 -t 02Dec09
```

これは、属性が (uuid1 OR uuid2) AND (02Dec09 以降) であるイベントを選択します。

オプション

次のオプションがサポートされています。

-A

ログの集約を実行します。コマンド行に 1 つ以上のログファイルパスが指定された場合は、それらのファイルを集約します。それ以外の場合は、すべての logadm ローテーションファイルを含む、すべての既知のログタイプを集約します。ログのマージは時間順に行われますが、その際、同じログファイル内の任意の 2 つのレコードは、集約内でも元のログファイル内とまったく同じ順番になります (これは、Fault Manager が受け取って処理した順番であり、基本的には時間順になる)。

ほかのログセット選択オプション -e、-i、-I を -A と併用することはできません。-c、-t、-T、-n などのフィルタオプションは使用できますが、-u は使用できません。出力オプション -v、-V、および -p は使用できますが、-m は使用できません。オプション -f は、選択されたすべてのログに追従します。

-c *class*

指定したクラスと一致するイベントを選択します。class 引数には、sh(1) に示されている構文と一致する glob パターンを使用できます。このクラスは、遠隔測定イベントの種類を示す階層分類文字列を表します。Sun の遠隔測定プロトコルの詳細については、Sun の Web サイト (http://www.oracle.com/us/sun/msg/) を参照してください。

-e

障害管理の障害ログではなくエラーログのイベントを表示します。

エラーログには、Sun の自動診断ソフトウェアが使用する非公開の遠隔測定情報が含まれています。この情報は、問題の事後分析やイベントの再現を支援するために記録されるものであるため、スクリプトやほかのツールの開発のために解析したり利用したりしないでください。非公開インタフェースに関する Sun の規則については、attributes(5) を参照してください。

-E *ENA*

指定された ENA 値に一致する任意の世代のイベントを選択します。ENA をサポートする検出機能でこのオプションを使用すると、同じ操作に関連付けられた複数のイベントを表示できます。

-f

ログファイル (-A 使用時は複数のログファイル) の増大に追従するため、追加データが発生するまで待機します。fmdump は、1 秒間休止し、ログファイルから

新しいデータを読み取って整形したあと、また休止状態に戻るという無限ループに入ります。このループは、割り込み (`Ctrl + C`) を送信することでいつでも終了できます。

`-m`
障害ログ内の各エントリに関連するローカライズされた診断メッセージを出力します。

`-n` *name*[.*name*]*[=*value*]
名前 (と場合によっては値) が一致するプロパティーを含むログイベントを (コマンド行で選択されたログから) 選択します。文字列プロパティーの場合は、この値を正規表現照合にできます。正規表現の構文は、`regex(5)` のマニュアルページの「EXTENDED REGULAR EXPRESSIONS」の節で説明されています。次の文字を使用する場合、

```
$  *  {  ^  |  (  )  \
```

または正規表現を使用する場合は、これらがシェルにとって意味を持つため、注意が必要です これらのすべてを単一引用符で囲むのがもっとも安全です。数値プロパティーの場合は、この値を 8 進数、16 進数、または 10 進数にすることができます。

`-p`
`-V` (非常に詳細) オプションと組み合わせた場合、使用可能な整形された出力オプションが実際に実行されることを要求します。

`-R` *dir*
`fmdump` がアクセスするログファイルのルートディレクトリとして、デフォルトのルート (/) ではなく指定したディレクトリを使用します。

`-t` *time*
指定した時間以降に発生したイベントを選択します。時間は次の形式で指定できます。

mm/*dd*/*yy* *hh*:*mm*:*ss*
月、日、年、時 (24 時間形式)、分、および秒。日付と時刻は任意の数のスペース文字で区切ることができます。シェルに 2 つの文字列を 1 つの引数として解釈させるには、引数全体を引用符で囲んでください。

mm/*dd*/*yy* *hh*:*mm*
月、日、年、時 (24 時間形式)、および分。日付と時刻は任意の数のスペース文字で区切ることができます。シェルに 2 つの文字列を 1 つの引数として解釈させるには、引数全体を引用符で囲んでください。

mm/*dd*/*yy*
指定した月、日、年の午前 12:00:00。

ddMonyy hh:*mm*:*ss*
日、月の名前、年、時 (24 時間形式)、分、および秒。日付と時刻は任意の数のスペース文字で区切ることができます。シェルに 2 つの文字列を 1 つの引数として解釈させるには、引数全体を引用符で囲んでください。

ddMonyy hh:*mm*
日、月の名前、年、時 (24 時間形式)、および分。日付と時刻は任意の数のスペース文字で区切ることができます。シェルに 2 つの文字列を 1 つの引数として解釈させるには、引数全体を引用符で囲んでください。

Mon dd hh:*mm*:*ss*
当年の月、日、時 (24 時間形式)、分、および秒。

yyyy-mm-dd `[T` *hh*:*mm*`[:`*ss*`]]`
年、月、日、省略可能な時 (24 時間形式)、分、および秒。秒、または時、分、秒を省略できます。

ddMonyy
指定した日、月の名前、年の午前 12:00:00。

hh:*mm*:*ss*
当日の時 (24 時間形式)、分、および秒。

hh:*mm*
当日の時 (24 時間形式) および分。

T`ns|`*T*`nsec`
T ナノ秒前。T は 10 進数で指定した整数です。

T`us |`*T*`usec`
T マイクロ秒前。T は 10 進数で指定した整数です。

T`ms|`*T*`msec`
T ミリ秒前。T は 10 進数で指定した整数です。

T`s | `*T*`sec`
T 秒前。*T* は 10 進数で指定した整数です。

T`m |`*T*`min`
T 分前。*T* は 10 進数で指定した整数です。

T`h |`*T*`hour`
T 時間前。*T* は 10 進数で指定した整数です。

T`d |`*T*`day`
T 日前。*T* は 10 進数で指定した整数です。

`-t` オプションの引数に `.`*n* の形式で小数を付加すると、指定する時間の小数部分を示すことができます。

-T *time*
指定した時間以前に発生したイベントを選択します。*time* は、-t オプションで説明したいずれかの時間形式を使って指定できます。

-u *uuid*
障害ログ内の問題診断イベントのうち、指定された *uuid* に厳密に一致するものを選択します。個々の診断には、識別用の汎用一意識別子 (UUID) が割り当てられています。-u オプションとほかのオプション (-v など) を組み合わせることにより、特定の診断に関連付けられたすべての詳細情報を表示できます。同じ問題診断 UUID に複数の障害ログイベントが関連付けられる可能性があります。単一の問題のライフサイクル (初期診断から最終解決まで) を記述するすべてのイベントが同じ問題 UUID を参照します。

-e オプションと -u オプションを同時に指定すると、指定した診断によって相互参照されるエラーイベントが表示されます。

-v
詳細なイベント情報を表示します。イベントの表示部分が拡大され、選択したイベントの追加の共通部分が表示されます。

-V
非常に詳細なイベント情報を表示します。イベントの表示部分が拡大され、各イベントに関連付けられたすべての名前-値ペアが表示されます。さらに、障害ログの場合は、イベントの表示部分に、診断に関連付けられていた該当するエラーの相互参照のリストも表示されます。

整形された出力を要求するには、-p と -V を併用します。

オペランド

次のオペランドがサポートされています。

file システム障害ログの代わりに表示する代替ログファイル (-A 使用時は複数の代替ログファイル) を指定します。fmdump ユーティリティーは、指定したログの種類を自動的に判別し、そのログに対応する適切な出力を生成します。

使用例

例 1 fmd ログからの特定のクラスの取得

fmd ログから、指定したクラスに関する情報を取得するには、次のいずれかのコマンドを使用します。完全なクラス名は ereport.io.ddi.context です。

```
# fmdump -Ve -c 'ereport.io.ddi.context'
# fmdump -Ve -c 'ereport.*.context'
# fmdump -Ve -n 'class=ereport.io.ddi.context'
# fmdump -Ve -n 'class=ereport.*.context'
```

前述のいずれのコマンドでも次の出力が生成されます。

```
Oct 06 2007 11:53:20.975021712 ereport.io.ddi.context
        nvlist version: 0
                class = ereport.io.ddi.context
                ena = 0x1b03a15ecf00001
```

例 1　fmd ログからの特定のクラスの取得　　(続き)

```
                detector = (embedded nvlist)
                nvlist version: 0
                        version = 0x0
                        scheme = dev
                        device-path = /
                (end detector)

                __ttl = 0x1
                __tod = 0x470706b0 0x3a1da690
```

例 2　fmd ログからの特定の検出機能デバイスパスの取得

次のコマンドは、fmd ログから検出機能デバイスパスを取得します。

```
# fmdump -Ve -n 'detector.device-path=.*/disk@1,0$'
Oct 06 2007 12:04:28.065660760 ereport.io.scsi.disk.rqs
nvlist version: 0
       class = ereport.io.scsi.disk.rqs
       ena = 0x453ff3732400401
       detector = (embedded nvlist)
                nvlist version: 0
                        version = 0x0
                        scheme = dev
                        device-path = /pci@0,0/pci1000,3060@3/disk@1,0
                (end detector)

                __ttl = 0x1
                __tod = 0x4707094c 0x3e9e758
```

終了ステータス

次の終了値が返されます。

0　正常終了。ログファイル内のすべてのレコードが検査され、問題はありませんでした。

1　致命的なエラーが発生。このため (指定されたファイルを開くのに失敗したなどの理由で)、ログファイルのデータは検査されませんでした。

2　無効なコマンド行オプションが指定されました。

3　ログファイルを開くことには成功しましたが、入出力エラーが発生したか、レコード自体の形式が不正だったため、ログファイルの 1 つ以上のレコードが表示されませんでした。fmdump は、表示できなかったレコードごとに警告メッセージを表示したあと、処理を続行し、ほかのレコードを表示しようとします。

ファイル

/var/fm/fmd	障害管理ログのディレクトリ
/var/fm/fmd/errlog	障害管理のエラーログ
/var/fm/fmd/fltlog	障害管理の障害ログ
/var/fm/fmd/infolog_hival	高価値情報ログ
/var/fm/fmd/infolog	情報ログ

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management
インタフェースの安定性	下記を参照。

コマンド行オプションは「不確実」です。人間が読める形式のエラーログおよび情報ログ出力は「非公開」です。人間が読める形式の障害ログ出力は「不確実」です。

関連項目

sh(1), fmadm(1M), fmd(1M), fmstat(1M), logadm(1M), syslogd(1M), libexacct(3LIB), attributes(5), regex(5)

『Oracle Solaris の管理: セキュリティーサービス』

http://www.oracle.com/us/sun/msg/

注意事項

障害ログには、エラーログに格納されたレコードへの参照が含まれています。fmdump -V を使ってこれらのレコードを表示することにより、特定の障害の診断に使用されたエラーを知ることができます。ログのローテーションによってエラーログの名前が変更されても、これらのリンクは保持されます。エラーログファイルを削除したり、エラーログファイルを別のファイルシステムのディレクトリに移動したりすると、これらのリンクが壊れることがあります。fmdump では、このような壊れたリンクのエラー情報を表示できません。障害ログに含まれる情報は引き続きすべて表示されます。

名前

fmstat – 障害管理モジュールの統計情報の報告

形式

`fmstat [-astTz] [-d u | d ] [-m module] [interval [count]]`

機能説明

fmstat ユーティリティーは、管理者とサービス担当者が Solaris Fault Manager (fmd(1M)) および関連する一連のモジュールに関連付けられた統計情報を報告するために使用できます。Fault Manager は各 Solaris システムのバックグラウンドで実行されます。Fault Manager は、システムソフトウェアが検出した問題に関する遠隔測定情報を受け取り、問題を診断し、障害のあるコンポーネントの無効化などの予測的自己修復操作を開始します。

fmstat を使って、障害管理に現在関与している診断エンジンおよびエージェントの統計情報を表示できます。fmd(1M)、fmadm(1M)、および fmdump(1M) の各ドキュメントには、障害管理操作を監視するツールに関する詳しい説明があります。

fmstat に -m オプションまたは -t オプションを指定すると、指定した障害管理モジュールに保持されている統計情報が報告されます。モジュールのリストは、fmadm config を使って取得できます。

fmstat に -m オプションを指定しないと、クライアントモジュールごとに次の統計情報が報告されます。

module	fmadm config で報告される障害管理モジュールの名前。
ev_recv	そのモジュールが受け取った遠隔測定イベントの数。
ev_acpt	そのモジュールが診断に必要なものとして受け入れたイベントの数。
wait	そのモジュールによる検査を待っている遠隔測定イベントの平均数。
svc_t	そのモジュールが受け取った遠隔測定イベントの平均サービス時間 (ミリ秒単位)。
%w	そのモジュールによる検査を待っている遠隔測定イベントが存在した時間の割合。
%b	そのモジュールが遠隔測定イベントを処理していた時間の割合。
open	そのモジュールが所有するアクティブなケース (未完了の問題調査) の数。
solve	そのモジュールが読み込み後に解決したケースの合計数。
memsz	そのモジュールによって現在割り当てられている動的メモリーの量。
bufsz	そのモジュールによって現在割り当てられている永続バッファー領域の量。

fmstat ユーティリティーを使用するユーザーは、SYS_ADMIN 特権を持っている必要があります。Solaris 特権を構成する方法の詳細については、『Oracle Solaris の管理: セキュリティーサービス』を参照してください。

オプション　次のオプションがサポートされています。

-a　fmd に保持されている情報も含めて、モジュールに関するすべての統計情報を出力します。-a オプションを指定しないと、モジュールに保持されている統計情報だけが報告されます。-m *module* を設定せずに -a オプションを使用すると、fmd に関連する一連のグローバル統計情報が表示されます。

-d u|d　タイムスタンプを表示します。

時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。 標準の日付フォーマットに d を指定します。date(1) を参照してください。

-m *module*　デフォルトの統計レポートの代わりに、指定した障害管理モジュールに関連する統計情報のレポートを出力します。モジュールは、Sun が障害管理ソフトウェア自体を保守するのに役立つ任意の統計情報を公開できます。モジュールの統計情報は、非公開インタフェースとなります。非公開インタフェースに関する Sun の規則については、attributes(5) を参照してください。障害管理モジュールの統計情報の値は予告なしに変更されることがあるため、これらの値に依存するスクリプトを作成しないでください。

-s　デフォルトのモジュール統計レポートの代わりに、モジュールに関連付けられた Soft Error Rate Discrimination (SERD) エンジンに関するレポートを出力します。SERD エンジンは、一定の時間 *T* におけるイベント数 *N* として測定された統計値がしきい値を超えたかどうかを判定するために障害管理ソフトウェアが使用する構成体です。-s オプションは、必ず -m オプションと組み合わせて使用します。

-t　個々の障害管理イベント転送に関連する統計情報のレポートを出力します。個々の障害管理モジュールは、1 つ以上のイベント転送の実装を提供できます。

-T　個々の障害管理イベント転送に関連する権限情報のテーブルを出力します。-m オプションを指定すると、指定したモジュールに関連する転送だけが表示されます。

-z　指定した障害管理モジュールに関するレポートから、値が 0 の統計情報を省略します。-z オプションは、必ず -m オプションと組み合わせて使用します。

オペランド　次のオペランドがサポートされています。

count　count レポートのみを出力して終了します。

interval　*interval* 秒ごとに新しいレポートを出力します。

interval と *count* を指定しないと、レポートを 1 つだけ出力して fmstat が終了します。*interval* を指定して *count* を指定しないと、fmstat はコマンドが中断されるまで *interval* 秒ごとにレポートを出力し続けます。

終了ステータス

次の終了値が返されます。

0 正常終了。

1 致命的なエラーが発生。致命的なエラーは、fmd(1M) との通信の失敗である可能性があります。また、要求された操作を実行するのに必要な特権が不十分だった可能性もあります。

2 無効なコマンド行オプションが指定されました。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management
インタフェースの安定性	下記を参照。

コマンド行オプションは「確実」です。人間が読めるデフォルトのレポートは不確実です。人間が読めるモジュールレポートは非公開です。

関連項目

fmadm(1M), fmd(1M), fmdump(1M), attributes(5)

『Oracle Solaris の管理: セキュリティーサービス』

名前

fmthard – ハードディスクへのラベルの設定

形式

SPARC

```
fmthard -d data | -n volume_name | -s datafile [-i] /dev/rdsk/c?
     [t?] d?s2
```

x86

```
fmthard -d data | -n volume_name | -s datafile [-i] /dev/rdsk/c?
     [t?] d?s2
```

機能説明

fmthard コマンドは、ハードディスクの VTOC (ボリューム構成テーブル) を更新し、さらに x86 システムでは Solaris fdisk パーティションにブート情報を追加します。ディスクラベルへの変更を要求するには、-s *datafile*、- d *data*、または -n *volume_name* のうち、1 つ以上のオプションを指定する必要があります。ディスクラベルの内容を出力する場合は、prtvtoc(1M) を参照してください。/dev/rdsk/c?[t?]d?s2 ファイルは、新しいラベルをインストールするデバイスの文字型特殊ファイルでなければなりません。x86 システムの場合、fmthard を使用する前に、ドライブに対して fdisk(1M) を実行する必要があります。

x86 システムを使用しているユーザーは、このページの「パーティション」という用語が、x86 マシン上の x86 fdisk パーティション内のスライスを指していることに注意してください。fmthard で作成されるパーティションと fdisk で作成されるパーティションを混同しないでください。

オプション

次のオプションがサポートされています。

-d *data*
: このオプションの *data* 引数は、現在の VTOC 内の特定のパーティションの情報を表す文字列です。この文字列の形式は *part:tag:flag:start:size* になっている必要があります。ここで、*part* はパーティション番号、*tag* はパーティションの ID タグ、*flag* は一連のアクセス権フラグ、*start* はパーティションの開始セクター番号、*size* はパーティション内のセクター数です。これらのフィールドの詳細については、後述の *datafile* の説明を参照してください。

-i
: このオプションを指定した場合、このコマンドは目的の VTOC テーブルを作成するものの、ディスクの VTOC を変更しないで標準出力に情報を出力します。

-n *volume_name*
: このオプションを使用すると、8 文字までの長さの *volume_name* がディスクに設定されます。

-s *datafile*
: このオプションを使用すると、ユーザーが作成した *datafile* に従って VTOC に情報が設定されます。*datafile* が – (ハイフン) の場合、fmthard は標準入力から情報を読み取ります。*datafile* の形式については後述します。このオプションでは、すべてのディスクパーティションのタイムスタンプフィールドがゼロに設定されます。

fmthard が生成するすべての VTOC には規則上、ディスク全体に対応するパーティション 2 も含まれます。*datafile* の入力にパーティション 2 のエントリが指定されていなかった場合、パーティション 2 のデフォルトエントリが VTOC 内に自動的に作成され、そのタグは V_BACKUP に、サイズはディスク全体のサイズに等しくなります。

datafile には、パーティション 0 から順に、パーティションごとに指定行が 1 つずつ含まれます。各行は復帰改行文字 (\n) で区切ります。行の先頭文字がアスタリスク (*) の場合、その行はコメント行とみなされます。各行は位置に依存するエントリからなり、スペースで区切ります。形式は次のとおりです。

partition tag flag starting_sector size_in_sectors

この場合、エントリには次の値が含まれます。

partition
: パーティション番号。Solaris SPARC では現在のところ、1 つのディスクに 0 から 7 までの最大 8 個のパーティションを作成できます。*partition* フィールドは 4 ビットですが、現時点では 3 ビットしか使用されていません。x86 の場合は 4 ビットすべてが使用されるため、0 から 15 までのスライスを使用できます。各 Solaris fdisk パーティションには 16 個までのスライスを作成できます。

tag
: パーティションタグ: 10 進数。予約済みのコードを次に示します。0 (V_UNASSIGNED)、1 (V_BOOT)、2 (V_ROOT)、3 (V_SWAP)、 4 (V_USR)、5 (V_BACKUP)、6 (V_STAND)、7 (V_VAR)、8 (V_HOME)、12 (V_SYSTEM)、および 24 (V_BIOS_BOOT)。

flag
: このフラグを使用すると、マウント不可能と読み取り専用のいずれかのフラグをパーティションに付けることができます。なお、マスクは V_UNMNT 0x01 と V_RONLY 0x10 です。マウント可能なパーティションでは 0x00 を使用します。

starting_sector
: パーティションの開始セクター番号 (10 進数)。

size_in_sectors
: パーティションによって占有されるセクターの数 (10 進数)。

prtvtoc コマンドの出力をファイルに保存して編集し、それを -s オプションの *datafile* 引数として使用できます。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

uname(1), format(1M), prtvtoc(1M), attributes(5)

x86 のみ

fdisk(1M), installgrub(1M)

注意事項

既存の VTOC を上書きする際には特に注意してください。不正なエントリが設定されると、現在のデータにアクセスできなくなる可能性があります。念のため、古い VTOC を保存してください。

ディスクの容量が 2 テラバイト未満の場合、fmthard は、ラベルの付いていないディスクに VTOC を書き込むことができません。そのような目的には format(1M) を使用してください。

名前

format – ディスクのパーティション分割および保守のためのユーティリティー

形式

```
format [-f command-file] [-l log-file] [-x data-file]
     [-d disk-name] [-t disk-type] [-p partition-name]
     [-s] [-m] [-M] [-e] [disk-list]

format -L label-type -d disk-name
```

機能説明

format を使用すると、システムディスクのフォーマット、ラベル付け、修復、および分析が行えます。従来のディスク保守用プログラムとは異なり、format は SunOS 環境で実行します。システムの稼働中はシステムディスクへの操作が制限されるので、format はメモリー常駐型のシステム環境でも使用することができます。しかし、大部分のアプリケーションでは、SunOS 環境で format を実行する方が簡単です。

-x オプションが指定されると、format は最初に *data-file* で定義されているディスクリストを使用します。次に format は、FORMAT_PATH 環境変数、コロンで区切られたファイル名またはディレクトリ (あるいはその両方) のリストを調べます。ディレクトリの場合、format はそのディレクトリで format.dat というファイルを検索します。ファイル名は絶対パス名でなければならず、そのまま使用されます。format は指定された各ファイルのディスクおよびパーティション定義をすべて作業用セットに追加します。複数の同じ定義は無視され、そのことが通知されることはありません。FORMAT_PATH が設定されていない場合、パスはデフォルトで /etc/format.dat になります。

disk-list は、c?t?d?、/dev/rdsk/c?t?d?s?、または /dev/chassis/?/disk 形式のディスクリストです。最後の 2 つの形式では、シェルのワイルドカード指定がサポートされています。たとえば、/dev/rdsk/c2* を指定すると、format はコントローラ c2 だけに接続されている全ドライブを操作対象とします。*disk-list* が指定されていない場合、format は、操作対象となり得る、システム上に存在するすべてのディスクを一覧表示します。

着脱式メディア装置がリストに含まれるのは、ユーザーがエキスパートモード (-e オプション) で format を実行した場合だけです。この機能は、下位互換性を確保するために用意されています。書き替え可能な着脱式メディア装置には rmformat(1) を使用します。

オプション

次のオプションがサポートされています。

-d *disk-name*	format プログラム開始時にカレントにするディスクを指定します。ディスクは、その論理名 (-d c0t1d0 や /dev/chassis/SYS/HD0/disk など) で指定します。ディスクリストにディスクを 1 つだけ指定しても、同じ結果が得られます。
-e	SCSI エキスパートメニューを有効にします。ただし、このオプションは、不用意に使用しないでください。

-f *command-file*	標準入力ではなく、*command-file* からコマンド入力を受け付けます。ファイルにはキーボードから入力した場合とまったく同じコマンドが指定されていなければなりません。このオプションが指定されている場合、format は continue? プロンプトを出力しません。*command-file* 内に y(es) または n(o) の応答を指定する必要はありません。非対話モードの場合、format はディスク選択番号の入力を求めることはありません。format を起動するときに、-d *disk-name* オプションを使用して現在の作業用ディスクを指定するか、または *command-file* に disk およびディスク選択番号を指定する必要があります。
-l *log-file*	指定された *log-file* に format セッションのトランスクリプトを記録します。これには標準入力、標準出力、および標準エラー出力が含まれます。
-L *label-type*	ただちに、かつ非対話形式でタイプ *label-type* のデフォルトラベルを、-d で指定されたディスクに書き込みます。*label-type* には efi または vtoc を指定する必要があります。既存のラベルがある場合は、*label-type* で上書きされます。x86 マシンでは、ディスク全体がデフォルトで、*label-type* のラベルが付いた 1 つの Solaris パーティションになり、すべての fdisk パーティションは失われます。
-m	拡張メッセージを有効にします。エラー発生時に詳細情報が得られます。
-M	拡張メッセージおよび診断メッセージを有効にします。フォーマット中に、 SCSI デバイスのモードページの状態についてさまざまな情報が得られます。
-p *partition-name*	プログラムの開始時にカレントにするディスクのパーティションテーブルを指定します。このテーブルは、データファイルで定義されている名前を使用して指定します。このオプションを使用できるのは、ディスクがカレントになっており、さらにディスクタイプが指定されているか、ディスクタイプがディスクラベルから取得できる場合に限られます。
-s	サイレント。あらゆる標準出力を抑制します。エラーメッセージは引き続き表示されます。このオプションは通常、-f オプションと組み合わせて使用します。
-t *disk-type*	プログラムの開始時にカレントにするディスクのタイプを指定します。ディスクタイプは、データファイルで定義されている名前を使用して指定します。このオプションを使用できるのは、前述のとおり、ディスクがカレントになっている場合に限られます。

-x *data-file* *data-file* に指定されているディスクのリストを使用します。

使用法

オプションを指定しないで format を起動した場合、あるいは、-e、-l、-m、-M、または -s オプションを指定して format を起動した場合、使用できるディスクを示した番号付きリストが表示され、リスト内の番号でディスクを指定するように求められます。マシンに 2 画面以上のディスクが搭載されている場合は、スペースキーを押して、次の 1 画面分のディスクを表示します。

ディスクが現在の画面に表示されていなくても、リスト内の番号でディスクを指定できます。たとえば、画面にディスク 11 ～ 20 が表示されている場合に 25 を入力すると、リスト内の 25 番目のディスクを指定できます。現在の画面に表示されていないディスクの番号を入力した場合は、選択の確認を求めるプロンプトが表示されます。表示リストに含まれている番号を入力した場合は、そのまま選択が受け入れられ、プロンプトは出力されません。

ディスクを指定すると、メインメニューが表示されます。このメニューでは次の作業を実行できます。

analyze
: 読み取り、書き込み、比較テスト、およびデータ消去を実行します。データ消去機能には、米国コンピュータセキュリティーセンター (National Computer Security Center、NCSC) の『A Guide to Understanding Data Remanence in Automated Information Systems』(NCSC-TG-025 バージョン 2) による上書きアルゴリズムが実装されています。「注意事項」を参照してください。

backup
: バックアップラベルを検索します。

cache
: 書き込みキャッシュおよび読み取りキャッシュを有効または無効にします。またはその状態を照会します。このメニュー項目が表示されるのは、-e オプションを指定して format を起動した場合だけです。このオプションを使用できるのは、SCSI デバイスに限られます。

current
: デバイス名、ディスクジオメトリ、およびディスク装置のパス名を表示します。

defect
: 欠陥領域リストを検索して出力します。このオプションを使用できるのは、SCSI デバイスに限られます。IDE ディスクの場合は、自動欠陥領域管理機能が実行されます。defect オプションを IDE ディスクで使用すると、次のメッセージが出力されます。

```
Controller does not support defect management
or disk supports automatic defect management.
```

disk
: 以降の操作で使用するディスク (カレントディスク) を選択します。

fdisk
: fdisk(1M) プログラムを実行し、Solaris ソフトウェア (x86 ベースのシステムのみ) 用の fdisk パーティションを作成します。

format　カレントディスクをフォーマットして検証します。このオプションを使用できるのは、SCSI デバイスに限られます。IDE ディスクは製造元により、あらかじめフォーマットされています。IDE ディスクに対して format オプションを使用すると、次のメッセージが出力されます。

```
Cannot format this drive. Please use your
manufacturer-supplied formatting utility.
```

inquiry　カレントドライブの製造元、製品名、およびリビジョンレベルを表示します。

label　カレントディスクに新しいラベルを書き込みます。

partition　スライスを作成または変更します。

quit　フォーマットメニューを終了します。

repair　ディスク上の特定のブロックを修復します。

save　新しいディスクおよびスライス情報を保存します。

type　ディスクタイプを選択 (定義) します。

verify　ラベルを読み取って表示します。シリンダ数、代替シリンダ数、ヘッド数、セクター数、パーティションテーブルなどの情報を出力します。

volname　8 文字の新しいボリューム名を用いて、ディスクにラベルを付けます。

環境変数

FORMAT_PATH　ディスクとパーティション定義用のコロンで区切られたファイル名またはディレクトリ (あるいはその両方)。ディレクトリを指定した場合、format はそのディレクトリでファイル format.dat を検索します。

ファイル

/etc/format.dat　デフォルトのデータファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

fmthard(1M), prtvtoc(1M), rmformat(1), format.dat(4), attributes(5), sd(7D)

『Oracle Solaris の管理: 一般的なタスク』

x86 のみ

fdisk(1M)

警告

format 機能を選択して Maxtor 207MB ディスクをフォーマットすると、次のメッセージが出力されます。

```
Mode sense page(4) reports rpm value as 0, adjusting it to 3600
```

これはドライバのバグであり、他社の旧式のドライブでも発生する可能性があります。前述のメッセージはエラーではありません。ドライブは正常に動作します。

シリンダ 0 にはパーティションテーブル (ディスクラベル) がありますが、raw ディスクパーティションで使用すると、他社のソフトウェアによって上書きされることがあります。

`format` は容量が 1 TB を超えるディスクまたは LUN をサポートするために、EFI に準拠したディスクラベルの記載をサポートします。しかし、ファイルシステムやボリュームマネージャなど、多くのソフトウェアコンポーネントは容量がまだ 1 TB もしくはそれ以下に制限されているので、注意が必要です。詳細は、『『Oracle Solaris の管理: 一般的なタスク』』を参照してください。

デフォルトでは、ディスクにラベルがない場合、2T バイトを超えるディスクには EFI ラベルが書き込まれます。`format` を `-e` オプションを指定して呼び出した場合は、ラベルの書き込み時にラベルの種類を選択できます。EFI ラベルが付いたディスク上でのブートは、現在サポートされていません。

注意事項

`format` にはヘルプ機能が用意されており、入力を求められたときにいつでも使用できます。要求されている情報についてヘルプが必要な場合は、単に疑問符 (?) を入力します。求められている情報についての簡単な説明が出力されます。メニュープロンプトに ? を入力すると、使用できるコマンドの一覧が表示されます。

SCSI ディスクの場合、Primary および Grown 両方の欠陥領域リストでフォーマットがデフォルトで実行されます。フォーマットを実行する前に、欠陥領域メニューで Primary リストだけを抽出しておくと、Primary リストに対するフォーマットだけが実行されます。

キャッシュの状態を変更できるのは、SCSI デバイスだけです。また、すべての SCSI デバイスがキャッシュ状態の変更または保存をサポートしているわけではありません。

上書きのための NCSC-TG-025 アルゴリズムは、DoD 5200.28-M (自動データ処理 (ADP) セキュリティ マニュアル) の「ERASE PROCEDURES」による仕様を満たしています。また、米国立標準技術研究所 (National Institute of Standards and Technology、NIST) による『Guidelines for Media Sanitization』(NIST SP 800-88) もこのアルゴリズムを参照しています。

名前

fpsd – Fp-scrubber daemon

形式

/usr/lib/fps/fpsd

機能説明

fpsd is a user-level daemon that periodically runs non-intrusive tests to validate proper functioning of FPU (Floating Point Unit) hardware in the system. A fault management action is initiated by means of fmd(1M), in case an error is detected by the test.

fpsd is not supported on all Sun systems. On a system on which it is not supported, fpsd is automatically disabled.

終了ステータス

The following exit values are returned:

0
: Successful completion.

non-zero
: An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/floating-point-scrubber
Interface Stability	Uncommitted

関連項目

svcs(1), fmd(1M), svcadm(1M), svccfg(1M), attributes(5), smf(5)

注意事項

The fpsd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/fpsd:default
```

You can use svccfg(1M) to change the default fpsd behavior:

```
Property Name            Type     Description
-------------            ----     -----------
config/exclude_cpus      astring  comma delimited list of
                                   CPU IDs to be excluded
                                   from proactive testing.
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

Disabling the service can increase system's vulnerability to silent data corruption, if a fault were to develop in an FPU within the system.

名前

fruadm – prints and updates customer data associated with FRUs

形式

```
/usr/platform/sun4u/sbin/fruadm

/usr/platform/sun4u/sbin/fruadm -l

/usr/platform/sun4u/sbin/fruadm [-r] path [text]
```

機能説明

`fruadm` prints or sets the customer data for Field-Replaceable Units (FRUs).

Without arguments, `fruadm` prints the paths of all FRU ID-capable FRUs (containers) in the system, along with the contents of the customer data record, if present, for each such FRU; for FRUs without customer data, `fruadm` prints only the container's path.

Only a privileged user can create or update data in containers. The privileges required to perform these write operations are hardware dependent. Typically, a default system configuration restricts write operations to the superuser or to the platform-administrator user.

オプション

The following options are supported:

`-l` List the system's frutree paths.

`-r` Recursively display or update the data for all containers rooted at the argument *path*.

オペランド

The following operands are supported:

path A full or partial system frutree path for or under which to print or set the customer data. The first field of each line of output of `fruadm -l` gives the valid full `frutree` paths for the system.

Paths can include shell meta-characters; such paths should be quoted appropriately for the user's shell. For partial paths, the first matching full path is selected for display or update. Without the `-r` option, the path must be that of a container; with the `-r` option, all containers (if any) under *path* will be selected.

text Up to 80 characters of text set as the customer data. If the text contains white space or shell metacharacters, it should be quoted appropriately for the user's shell.

使用例

例 1 Displaying All Customer Data

The following example prints all customer data available from `FRU`s on the system. For containers with no customer data, only the containers' paths will be listed.

```
example% fruadm
```

例 2 Displaying Customer Data For a Single FRU

The following command prints the customer data, if present, for the specified FRU:

```
example% fruadm /frutree/chassis/system-board
```

例 3 Displaying Customer Data For a Single FRU

The following command prints the customer data, if present, for the first `mem-module` found:

```
example% fruadm mem-module
```

例 4 Setting Customer Data

The following example sets the customer data for a FRU:

```
example# fruadm system-board 'Asset Tag 123456'
```

例 5 Setting Customer Data

The following command sets the customer data for all FRUs under `chassis`:

```
example# fruadm -r /frutree/chassis "Property of XYZ, Inc."
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/fru-id
Interface Stability	Uncommitted

関連項目

`prtfru(1M)`, `attributes(5)`

名前

fsck – ファイルシステムの検査および修復

形式

```
fsck [-F FSType] [-m] [-V] [-v] [special]...

fsck [-F FSType] [-n | N | y | Y] [-V] [-v]
     [-o FSType-specific-options] [special]...
```

機能説明

fsck はファイルシステムの状態を検査し、不整合を対話形式で修復します。ファイルシステムに不整合がある場合、デフォルトでは、修復処理を行う前に、ユーザーからの yes または no の応答を待機します。ユーザーに書き込み権が与えられていない場合、fsck はデフォルトで no の応答に対する動作をします。修復処理によっては、データが失われることがあります。データが失われる量と重要度は、診断出力から判断できます。

FSType-specific-options は、(スペースを入れずに) コンマで区切ったオプションのリストまたはキーワード / 属性のペアのリストとして指定します。これらは、*FSType*-固有のコマンドモジュールによって解釈されます。

special は、ファイルシステムが配置されている文字型特殊デバイスを指定します。/dev/rdsk/c1t0d0s7 などがその例です。注 : ブロック型特殊デバイスではなく、文字型特殊デバイスを使用してください。fsck は、ブロック型特殊デバイスがマウントされている場合には、動作しません。

special デバイスを指定しなかった場合、fsck は /etc/vfstab に指定されているファイルシステムを検査します。/etc/vfstab の中で、fsckdev フィールドに文字型特殊デバイスエントリがあり、fsckpass フィールドにゼロ以外の数値エントリのあるエントリが検査されます。-F *FSType* を指定すると、指示されたタイプのファイルシステムタイプだけが検査されます。

special が指定されていても、 -F が指定されていない場合は、/etc/vfstab の対応するエントリを探すことによって、ファイルシステムタイプが判別されます。対応するエントリがない場合は、/etc/default/fs に指定されているデフォルトのローカルファイルシステムタイプが使用されます。

ファイルシステムタイプが並列検査をサポートしている場合 (たとえば ufs) 、検査可能ないくつかのファイルシステムを並列に検査できます。詳細は、ファイルシステム固有のマニュアルページ (fsck_ufs(1M) など) を参照してください。

オプション

以下の汎用オプションがサポートされています。

-F *FSType*
: 操作するファイルシステムのタイプを指定します。

-m
: 検査だけで修復は行いません。このオプションを指定すると、ファイルシステムがマウントできるステータスかどうかが検査され、該当する終了ステータスが返されます。ファイルシステムがマウントできる状態であれば、fsck は次のメッセージを表示します。

```
ufs fsck: sanity check: /dev/rdsk/c0t3d0s1 okay
```

-n | -N
fsck からのすべての問い合わせに対して no の応答をするものとみなします。ファイルシステムを書き込みでオープンしません。

-V
コマンド行を展開して表示しますが、コマンドは実行しません。このオプションは、コマンド行の検査および検証するのに使用します。

-v
詳細な出力を有効にします。一部のファイルシステム固有の fsck 実装では、サポートされない可能性があります。

-y | Y
fsck からのすべての問い合わせに対して yes の応答をするとみなします。

-o *specific-options*
specific-options には、次のサブオプションの組み合わせを (スペースは入れずに) コンマで区切って指定します。

b=*n*
ファイルシステムのスーパーブロックとしてブロック *n* を使用します。ブロック 32 はつねに、代替スーパーブロックの 1 つです。-Nv オプションを指定して newfs(1M) を実行すると、他のスーパーブロックの位置を調べることができます。

c
古い (静的テーブル) 形式のファイルシステムを新しい (動的テーブル) 形式に変換します。ファイルシステムが新しい形式のときは、ファイルシステム構成が古い形式をサポートできる場合、古い形式に変換します。対話モードの場合、変換の向きが表示され、変換を実行するかどうかが尋ねられます。否定応答を返すと、そのファイルシステムに対してそれ以上の操作は実行されません。非対話モードの場合は、変換の向きが表示され、可能であれば、ユーザーとの対話なしで実行されます。すべてのファイルシステムを一括変換する場合は、非対話モードでの変換が便利です。ファイルシステムのタイプは fstyp(1M) の出力の先頭行からわかります。注 : c オプションを使用することはほとんどありません。4.1 より前のリリースとの互換性を確保する場合に限って指定してください。今後のリリースにこのオプションが含まれるかどうかは保証されません。

f
スーパーブロックのクリーンフラグの状態に関係なく、ファイルシステムを強制的に検査します。

p
非対話 (preen) モードでファイルシステムを検査して修復します。ユーザーとの対話を必要とする問題が検出された場合には、ただちに終了します。並列ファイルシステム検査を有効にする場合、このオプションは必須です。

w
書き込み可能なファイルシステムだけを検査します。

終了ステータス

0
ファイルシステムがマウント解除されており、問題なし

1
誤ったパラメータが指定された

32
ファイルシステムがマウント解除されており、検査が必要 (fsck -m オプションの場合のみ)

33
ファイルシステムはマウント済み

34
デバイスの状態が取得できない

35
読み取り/書き込み用にマウントされたファイルシステムが変更された - リブート

36
修正不可能なエラーが検出された - 通常どおりに終了

37
処理中にシグナルが捕捉された

39
修正不可能なエラーが検出された - 即時終了

40
ファイルシステムは読み取り専用でマウントされており、問題なし

使用法

fsck コマンドは、largefile(5) のマニュアルページに従い、UFS ファイルシステムの大規模ファイルを認識します。

ファイル

/etc/default/fs
デフォルトのローカルのファイルシステムタイプ。デフォルト値は、/etc/default/fs 内で次のように設定されています。例: LOCAL=ufs。

LOCAL
FSType が指定されていない場合はコマンドのデフォルトファイルタイプ

/etc/vfstab
各ファイルシステム用のデフォルトのパラメータリスト

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

clri(1M), fsck_ufs(1M), fsdb_ufs(1M), fsirand(1M), fstyp(1M), mkfs(1M), mkfs_ufs(1M), mountall(1M), newfs(1M), reboot(1M), vfstab(4), attributes(5), largefile(5), ufs(7FS)

警告

オペレーティングシステムはファイルシステムデータをバッファリングします。マウントされているファイルシステムに fsck を実行すると、オペレーティングシステムのバッファー内容がディスク上のものよりも古くなります。そのため、fsck を使用する場合は、ファイルシステムをマウント解除してください。それができない場合は、fsck の実行後に、システムを停止させ、ただちにリブートしてください。それでも不十分な場合がしばしばあります。ファイルシステム上で fsck を実行した結果、ファイルシステムが変更された場合には、パニックが発生することがあります。

注意事項

このコマンドは、すべての *FSTypes* で使用できるわけではありません。

Solaris 9 以降、fsck はディスク上の拡張属性データを管理します (拡張ファイル属性については、fsattr(5) を参照)。 拡張属性が設定されたファイルシステムを、属性を認識しないバージョンの Solaris (Solaris 9 より前のバージョン) にマウントすることはできますが、属性にアクセスすることはできません。fsck はファイルからそれらの属性を取り除き、lost+found に移します。属性が取り除かれたファイルシステムは、属性を認識するバージョンの Solaris 上でも完全に安定動作しますが、破壊しているファイルシステムとみなされてしまいます。この場合、属性を認識するバージョンの fsck を実行し、ファイルシステムを安定させてから、属性を認識する環境でそのファイルシステムを使用してください。

名前

fsck_pcfs – file system consistency check and interactive repair

形式

`fsck -F pcfs [generic_options] special`

`fsck -F pcfs [generic_options] [-o specific_options] special`

機能説明

The `fsck` utility audits and interactively repairs inconsistent conditions on file systems. *special* represents the character special device on which the file system resides, for example `/dev/disk`. The character special device, not the block special device, should be used.

In the case of correcting serious inconsistencies, by default, `fsck` asks for confirmation before making a repair and waits for the operator to respond either `yes` or `no`. If the operator does not have write permission on the file system, `fsck` defaults to a `-n` (no corrections) action. See `fsck(1M)`.

Repairing some file system inconsistencies may result in loss of data. The amount and severity of data loss may be determined from the diagnostic output.

When executed with the verify option (`-o v`), `fsck_pcfs` automatically scans the entire file system to verify that all of its allocation units are accessible. If it finds any units inaccessible, it updates the file allocation table (FAT) appropriately. It also updates any effected directory entries to reflect the problem. This directory update includes truncating the file at the point in its allocation chain where the file data is no longer accessible. Any remaining accessible allocation units become orphaned.

Orphaned chains of accessible allocation units are, with the operator's concurrence, linked back into the file system as files in the root directory. These files are assigned names of the form file*NNNN*.chk, where the *N*s are digits in the integral range from `0` through `9`.

After successfully scanning and correcting any errors in the file system, `fsck` displays a summary of information about the file system. This summary includes the size of the file system in bytes, the number of bytes used in directories and individual files, and the number of available allocation units remaining in the file system.

オプション

generic_options

The following generic options are supported:

`-m`

Check but do not repair. This option checks that the file system is suitable for mounting, returning the appropriate exit status. If the file system is ready for mounting, `fsck` displays a message such as:

```
pcfs fsck: sanity check:
/dev/disk okay
```

`-n | -N`

Assume a no response to all questions asked by `fsck`; do not open the file system for writing.

`-V`

Echo the expanded command line, but do not execute the command. This option may be used to verify and to validate the command line.

-y | -Y Assume a yes response to all questions asked by fsck.

-o *specific_options* Specify pcfs file system specific options in a comma-separated list, in any combination, with no intervening spaces.

v Verify all allocation units are accessible prior to correcting inconsistencies in the metadata.

p Check and fix the file system non-interactively (preen). Exit immediately if there is a problem requiring intervention.

w Check writable file systems only.

ファイル

special The device which contains the pcfs. A hard disk device or high-capacity removable device name much be qualified with a suffix to indicate the proper FDISK partition.

For example, in the names: /dev/rdsk/c0t0d0p0:c and /dev/rdsk/c0t4d0s2:c, the :c suffix indicates the first partition on the disk contains the pcfs.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/pcfs
Interface Stability	Committed

関連項目

fsck(1M), fstyp(1M), fdisk(1M), mkfs(1M), mkfs_pcfs(1M), mountall(1M), attributes(5), pcfs(7FS),

警告

The operating system buffers file system data. Running fsck on a mounted file system can cause the operating system's buffers to become out of date with respect to the disk. For this reason, the file system should be unmounted when fsck is used. If this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately after fsck is run. Quite often, however, this is not sufficient. A panic will probably occur if running fsck on a file system modifies the file system.

名前

fsck_udfs – file system consistency check and interactive repair

形式

```
fsck -F udfs [generic_options] [special ...]
```

```
fsck -F udfs [generic_options] [-o specific_options]
     [special ...]
```

機能説明

`fsck` audits and interactively repairs inconsistent conditions on file systems. A file system to be checked can be specified by giving the name of the block or character special device or by giving the name of its mount point if a matching entry exists in `/etc/vfstab`.

special represents the character special device, for example, `/dev/rdsk/c0t2d0s0`, on which the file system resides. The character special device, not the block special device should be used. `fsck` does not work on a mounted block device.

If no special device is specified, all `udfs` file systems specified in the `vfstab` file with a `fsckdev` entry are checked. If the `-p` (`preen`) option is specified, `udfs` file systems with an `fsckpass` number greater than `1` are checked in parallel. See `fsck`(1M).

In the case of correcting serious inconsistencies, by default, `fsck` asks for confirmation before making a repair and waits for the operator to respond with either `yes` or `no`. If the operator does not have write permission on the file system, `fsck` defaults to the `-n` (no corrections) option. See `fsck`(1M).

Repairing some file system inconsistencies can result in loss of data. The amount and severity of data loss can be determined from the diagnostic output.

`fsck` automatically corrects innocuous inconsistencies. It displays a message for each corrected inconsistency that identifies the nature of the correction which took place on the file system. After successfully correcting a file system, `fsck` prints the number of files on that file system and the number of used and free blocks.

Inconsistencies checked are as follows:

- Blocks claimed by more than one file or the free list
- Blocks claimed by a file or the free list outside the range of the file system
- Incorrect link counts in file entries
- Incorrect directory sizes
- Bad file entry format
- Blocks not accounted for anywhere
- Directory checks, file pointing to unallocated file entry and absence of a parent directory entry
- Descriptor checks, more blocks for files than there are in the file system
- Bad free block list format
- Total free block count incorrect

オプション

The following options are supported:

generic_options The following *generic_options* are supported:

`-m` Check but do not repair. This option checks to be sure that the file system is suitable for mounting, and returns the appropriate exit status. If the file system is ready for mounting, `fsck` displays a message such as:

```
udfs fsck: sanity check: /dev/rdsk/c0t2d0s0 okay
```

`-n | -N` Assume a `no` response to all questions asked by `fsck`; do not open the file system for writing.

`-V` Echo the expanded command line, but do not execute the command. This option can be used to verify and to validate the command line.

`-y | -Y` Assume a `yes` response to all questions asked by `fsck`.

`-o` *specific_options* Specify `udfs` file system specific options in a comma-separated list with no intervening spaces. The following *specific_options* are available:

`f` Force checking of file systems regardless of the state of their logical volume integrity state.

`p` Check and fix the file system non-interactively (`preen`). Exit immediately if there is a problem that requires intervention. This option is required to enable parallel file system checking.

`w` Check writable file systems only.

ファイル

`/etc/vtstab` List of default parameters for each file system.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/udfs

関連項目

`fsck(1M)`, `fsdb_udfs(1M)`, `fstyp(1M)`, `mkfs(1M)`, `mkfs_udfs(1M)`, `mountall(1M)`, `reboot(1M)`, `vfstab(4)`, `attributes(5)`

警告

The operating system buffers file system data. Running `fsck` on a mounted file system can cause the operating system's buffers to become out of date with respect to the disk. For this reason, use `fsck` only when the file system is unmounted. If this is not possible, take care that the system is quiescent and that it is rebooted immediately after running `fsck`. A panic will probably occur if running `fsck` on a file system that modifies the file system while it is mounted.

If an unmount of the file system is not done before the system is shut down, the file system might become corrupted. In this case, a file system check needs to be completed before the next mount operation.

診断

not writable
: You cannot write to the device.

Currently Mounted on
: The device is already mounted and cannot run `fsck`.

FILE SYSTEM WAS MODIFIED
: File system has been modified to bring it to a consistent state.

Can't read allocation extent
: Cannot read the block containing allocation extent.

Bad tag on alloc extent
: Invalid tag detected when expecting an allocation extent.

Volume sequence tag error
: Invalid tag detected in the volume sequence.

Space bitmap tag error
: Invalid tag detected in the space bitmap.

UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY
: Use `fsck` in interactive mode.

名前

fsck_ufs – file system consistency check and interactive repair

形式

```
fsck -F ufs [generic-options] [special]...

fsck -F ufs [generic-options] [-o specific-options]
      [special]...
```

機能説明

The `fsck` utility audits and interactively repairs inconsistent conditions on file systems. A file system to be checked may be specified by giving the name of the block or character *special* device or by giving the name of its mount point if a matching entry exists in `/etc/vfstab`.

The *special* parameter represents the character special device, for example, `/dev/rdsk/c1t0d0s7`, on which the file system resides. The character special device, not the block special device should be used. The `fsck` utility will not work if the block device is mounted, unless the file system is error-locked.

If no *special* device is specified, all `ufs` file systems specified in the `vfstab` with a `fsckdev` entry will be checked. If the `-p` (“preen") option is specified, `ufs` file systems with an `fsckpass` number greater than `1` are checked in parallel. See `fsck(1M)`.

In the case of correcting serious inconsistencies, by default, `fsck` asks for confirmation before making a repair and waits for the operator to respond either `yes` or `no`. If the operator does not have write permission on the file system, `fsck` will default to a `-n` (no corrections) action. See `fsck(1M)`.

Repairing some file system inconsistencies can result in loss of data. The amount and severity of data loss can be determined from the diagnostic output.

The `fsck` utility automatically corrects innocuous inconsistencies such as unreferenced inodes, too-large link counts in inodes, missing blocks in the free list, blocks appearing in the free list and also in files, or incorrect counts in the super block. It displays a message for each inconsistency corrected that identifies the nature of the correction on the file system which took place. After successfully correcting a file system, `fsck` prints the number of files on that file system, the number of used and free blocks, and the percentage of fragmentation.

Inconsistencies checked include:

- Blocks claimed by more than one inode or the free list.
- Blocks claimed by an inode or the free list outside the range of the file system.
- Incorrect link counts.
- Incorrect directory sizes.
- Bad inode format.
- Blocks not accounted for anywhere.
- Directory checks, file pointing to unallocated inode, inode number out of range, and absence of ‘.' and ‘.. ' as the first two entries in each directory.
- Super Block checks: more blocks for inodes than there are in the file system.

- Bad free block list format.
- Total free block and/or free inode count incorrect.

Orphaned files and directories (allocated but unreferenced) are, with the operator's concurrence, reconnected by placing them in the `lost+found` directory. The name assigned is the inode number. If the `lost+found` directory does not exist, it is created. If there is insufficient space in the `lost+found` directory, its size is increased.

An attempt to mount a `ufs` file system with the `-o nolargefiles` option will fail if the file system has ever contained a large file (a file whose size is greater than or equal to 2 Gbyte). Invoking `fsck` resets the file system state if no large files are present in the file system. A successful mount of the file system after invoking `fsck` indicates the absence of large files in the file system. An unsuccessful mount attempt indicates the presence of at least one large file. See `mount_ufs(1M)`.

オプション

The *generic-options* consist of the following options:

`-m` Check but do not repair. This option checks that the file system is suitable for mounting, returning the appropriate exit status. If the file system is ready for mounting, `fsck` displays a message such as:

```
ufs fsck: sanity check: /dev/rdsk/c0t3d0s1 okay
```

`-n | N` Assume a `no` response to all questions asked by `fsck`; do not open the file system for writing.

`-V` Echo the expanded command line, but do not execute the command. This option may be used to verify and to validate the command line.

`-v` Enables verbose output. Might not be supported by all filesystem-specific `fsck` implementations.

`-y | Y` Assume a `yes` response to all questions asked by `fsck`.

See generic `fsck(1M)` for the details for specifying *special*.

`-o` *specific-options* Specify `ufs` file system specific options. These options can be any combination of the following separated by commas (with no intervening spaces).

`b=`*n* Use block *n* as the super block for the file system. Block 32 is always one of the alternate super blocks. Determine the location of other super blocks by running `newfs(1M)` with the `-Nv` options specified.

`f` Force checking of file systems regardless of the state of their super block clean flag.

p Check and fix the file system non-interactively ("preen"). Exit immediately if there is a problem requiring intervention. This option is required to enable parallel file system checking.

w Check writable file systems only.

ファイル

`/etc/vfstab` list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`clri(1M)`, `fsck(1M)`, `fsdb_ufs(1M)`, `fsirand(1M)`, `fstyp(1M)`, `mkfs(1M)`, `mkfs_ufs(1M)`, `mount_ufs(1M)`, `mountall(1M)`, `newfs(1M)`, `reboot(1M)`, `vfstab(4)`, `attributes(5)`, `largefile(5)`, `ufs(7FS)`

警告

The operating system buffers file system data. Running `fsck` on a mounted file system can cause the operating system's buffers to become out of date with respect to the disk. For this reason, the file system should be *unmounted* when `fsck` is used. If this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately after `fsck` is run. Quite often, however, this will not be sufficient. A panic will probably occur if running `fsck` on a file system modifies the file system.

注意事項

It is usually faster to check the character special device than the block special device.

Running `fsck` on file systems larger than 2 Gb fails if the user chooses to use the block interface to the device:

```
fsck /dev/dsk/c?t?d?s?
```

rather than the raw (character special) device:

```
fsck /dev/rdsk/c?t?d?s?
```

名前

fsdb – file system debugger

形式

`fsdb [-F` *FSType*`] [-V] [-o` *FSType-specific_options*`]` *special*

機能説明

`fsdb` is a file system debugger that allows for the manual repair of a file system after a crash. *special* is a special device used to indicate the file system to be debugged. `fsdb` is intended for experienced users only. *FSType* is the file system type to be debugged. Since different *FSTypes* have different structures and hence different debugging capabilities, the manual pages for the *FSType*-specific `fsdb` should be consulted for a more detailed description of the debugging capabilities.

オプション

`-F` Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by matching the *special* with an entry in the table, or by consulting `/etc/default/fs`.

`-V` Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from `/etc/vfstab`. This option may be used to verify and validate the command line.

`-o` Specify *FSType*-specific options.

使用法

See `largefile(5)` for the description of the behavior of `fsdb` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/default/fs` default local file system type. Default values can be set for the following flags in `/etc/default/fs`. For example: LOCAL=ufs

`LOCAL:` The default partition for a command if no FSType is specified.

`/etc/vfstab` list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`vfstab(4)`, `attributes(5)`, `largefile(5)` Manual pages for the FSType-specific modules of `fsdb`.

注意事項

This command may not be supported for all *FSTypes*.

名前

fsdb_udfs – udfs file system debugger

形式

`fsdb [-F] udfs [generic_option] [-o specific_option] special`

機能説明

The `fsdb_udfs` command is an interactive tool that can be used to patch up a damaged `udfs` file system. `fsdb_udfs` has conversions to translate block and i-numbers into their corresponding disk addresses. Mnemonic offsets to access different parts of an inode are also included. Mnemonic offsets greatly simplify the process of correcting control block entries or descending the file system tree.

`fsdb` contains several error-checking routines to verify inode and block addresses. These can be disabled if necessary by invoking `fsdb` with the `-o` option or by using the `o` command.

`fsdb` reads one block at a time, and therefore works with raw as well as block I/O devices. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block. In order to modify any portion of the disk, `fsdb` must be invoked with the `-w` option.

Wherever possible, `adb`-like syntax has been adopted to promote the use of `fsdb` through familiarity.

オプション

The following options are supported:

`-o` *specific_option*

Specify `udfs` file system specific options in a comma-separated list with no intervening spaces. The following specific options are supported:

o	Override some error conditions.
p=*string*	Set prompt to *string*.
w	Open for write.
?	Display usage.

使用法

Numbers are considered hexadecimal by default. The user has control over how data is to be displayed or accepted. The `base` command displays or sets the input and output base. Once set, all input defaults to this base and all output displays in this base. The base can be overriden temporarily for input by preceding hexadecimal numbers by `0x`, preceding decimal numbers with a `0t`, or octal numbers with a `0`. Hexadecimal numbers beginning with `a-f` or `A-F` must be preceded with a `0x` to distinguish them from commands.

Disk addressing by `fsdb` is at the byte level. However, `fsdb` offers many commands to convert a desired inode, directory entry, block, and so forth, to a byte address. After the address has been calculated, `fsdb` records the result in the current address (`dot`).

Several global values are maintained by `fsdb`:

- Current base (referred to as `base`)
- Current address (referred to as `dot`)

- Current inode (referred to as `inode`)
- Current count (referred to as `count`)
- Current type (referred to as `type`)

Most commands use the preset value of `dot` in their execution. For example,

```
> 2:inode
```

first sets the value of dot (`.`) to `2`, colon (`:`), signifies the start of a command, and the `inode` command sets `inode` to `2`. A count is specified after a comma (`,`). Once set, count remains at this value until a new command is encountered that resets the value back to `1` (the default).

So, if

```
> 2000,400/X
```

is entered, `400` hex longs are listed from `2000`, and when completed, the value of dot is `2000` + `400` * `sizeof` (long). If a RETURN is then entered, the output routine uses the current values of `dot`, `count`, and `type` and displays `400` more hex longs. An asterisk (`*`) causes the entire block to be displayed. An example showing several commands and the use of RETURN would be:

```
> 2:ino; 0:dir?d
```

or

```
> 2:ino; 0:db:block?d
```

The two examples are synonymous for getting to the first directory entry of the root of the file system. Once there, subsequently entering a RETURN, plus (`+`), or minus (`-`) advances to subsequent entries. Notice that

```
> 2:inode; :ls
```

or

```
> :ls /
```

is again synonymous.

Expressions

The following symbols are recognized by `fsdb`:

RETURN — Update the value of dot by the current value of *type* and *display* using the current value of *count*.

— Update the value of dot by specifying a numeric expression. Specify numeric expressions using addition, subtraction, mulitiplication, and division operators (`+`, `-`, `*`, and `%`). Numeric expressions are evaluated from left to right and can use parentheses. After evaluation, the value of dot is updated.

, *count*	Update the count indicator. The global value of *count* is updated to *count*. The value of *count* remains until a new command is run. A *count* specifier of * attempts to show a blocks's worth of information. The default for *count* is `1`.
? *f*	Display in structured style with format specifier `f`. See `Formatted Output`.
/ *f*	Display in unstructured style with format specifier `f`. See `Formatted Output`.
.	Display the value of dot.
+*e*	Increment the value of dot by the expression *e*. The amount actually incremented is dependent on the size of type: `dot = dot + e * sizeof` (*type*) The default for *e* is `1`.
−*e*	Decrement the value of dot by the expression *e*. See `+`.
**e*	Multiply the value of dot by the expression *e*. Multiplication and division don't use *type*. In the above calculation of dot, consider the `sizeof` (*type*) to be `1`.
%*e*	Divide the value of dot by the expression *e*. See *.
< *name*	Restore an address saved in register *name*. *name* must be a single letter or digit.
> *name*	Save an address in register *name*. *name* must be a single letter or digit.
= *f*	Display indicator. If *f* is a legitimate format specifier (see `Formatted Output`), then the value of dot is displayed using format specifier *f*. Otherwise, assignment is assumed. See `= [s] [e]`.
= [*s*] [*e*]	Change the value of dot using an assignment indicator. The address pointed to by dot has its contents changed to the value of the expression *e* or to the ASCII representation of the quoted (") string *s*. This can be useful for changing directory names or ASCII file information.
=+ *e*	Change the value of dot using an incremental assignment. The address pointed to by dot has its contents incremented by expression *e*.
=- e	Change the value of dot using a decremental assignment. Decrement the contents of the address pointed to by dot by expression *e*.

Commands

A command must be prefixed by a colon (`:`). Only enough letters of the command to uniquely distinguish it are needed. Multiple commands can be entered on one line by separating them by a SPACE, TAB, or semicolon (`;`).

To view a potentially unmounted disk in a reasonable manner, `fsdb` supports the `cd`, `pwd`, `ls`, and `find` commands. The functionality of each of these commands basically matches that of its UNIX counterpart. See `cd(1)`, `pwd(1)`,`ls(1)`, and`find(1)` for details. The *, ,, ?, and - wildcard characters are also supported.

The following commands are supported:

base[=*b*]	Display or set the base. All input and output is governed by the current base. Without the = *b*, displays the current base. Otherwise, sets the current base to *b*. Base is interpreted using the old value of base, so to ensure correctness use the `0`, `0t`, or `0x` prefix when changing the base. The default for base is hexadecimal.
block	Convert the value of dot to a block address.
cd [*dir*]	Change the current directory to directory *dir*. The current values of inode and dot are also updated. If `dir` is not specified, changes directories to inode 2, root (`/`).
directory	If the current inode is a directory, converts the value of dot to a directory slot offset in that directory, and dot now points to this entry.
file	Set the value of dot as a relative block count from the beginning of the file. The value of dot is updated to the first byte of this block.
find *dir* [`-name` *n*] | [`-inum` *i*]	Find files by name or i-number. Recursively searches directory *dir* and below for file names whose i-number matches `i` or whose name matches pattern *n*. Only one of the two options (`-name` or `-inum`) can be used at one time. The find `-print` is not necessary or accepted.
fill=*p*	Fill an area of disk with pattern *p*. The area of disk is delimited by dot and count.
inode	Convert the value of dot to an inode address. If successful, the current value of inode is updated as well as the value of dot. As a convenient shorthand, if `:inode` appears at the beginning of the line, the value of dot is set to the current inode and that inode is displayed in inode format.
ls [`-R`] [`-l`] *pat1 pat2*...	List directories or files. If no file is specified, the current directory is assumed. Either or both of the options can be used (but, if used, must be specified before the filename specifiers). Wild card characters are available and multiple arguments are acceptable. The long listing shows only the i-number and the name; use the inode command with `?i` to get more information.
override	Toggle the value of override. Some error conditions might be overridden if override is toggled to `on`.
prompt "*p*"	Change the `fsdb` prompt to *p*. *p* must be enclosed in quotes.

pwd	Display the current working directory.
quit	Quit `fsdb`.
tag	Convert the value of dot and if this is a valid tag, print the volume structure according to the tag.
!	Escape to the shell.

Inode Commands

In addition to the above commands, several other commands deal with inode fields and operate directly on the current inode (they still require the colon (`:`). They can be used to more easily display or change the particular fields. The value of dot is only used by the `:db` and `:ib` commands. Upon completion of the command, the value of dot is changed so that it points to that particular field. For example,

```
> :ln=+1
```

increments the link count of the current inode and sets the value of dot to the address of the link count field.

The following inode commands are supported:

at	Access time
bs	Block size
ct	Creation time
gid	Group id
ln	Link number
mt	Modification time
md	Mode
maj	Major device number
min	Minor device number
nm	This command actually operates on the directory name field. Once poised at the desired directory entry (using the `directory` command), this command allows you to change or display the directory name. For example, `> 7:dir:nm="foo"` gets the 7th directory entry of the current inode and changes its name to `foo`. Directory names cannot be made larger than the field allows. If an attempt is made to make a directory name larger than the field allows,, the string is truncated to fit and a warning message is displayed.
sz	File size

uid User ID

uniq Unique ID

Formatted Output

Formatted output comes in two styles and many format types. The two styles of formatted output are: structured and unstructured. Structured output is used to display inodes, directories, and so forth. Unstructured output displays raw data.

Format specifiers are preceded by the slash (`/`) or question mark (`?`) character. *type* is updated as necessary upon completion.

The following format specifiers are preceded by the `?` character:

i Display as inodes in the current base.

d Display as directories in the current base.

The following format specifiers are preceded by the `/` character:

b Display as bytes in the current base.

c Display as characters.

o | O Display as octal shorts or longs.

d | D Display as decimal shorts or longs.

x | X Display as hexadecimal shorts or longs.

使用例

例 1 Using fsdb as a calculator for complex arithmetic

The following command displays `2010` in decimal format, and is an example of using `fsdb` as a calculator for complex arithmetic.

```
> 2000+400%(20+20)=D
```

例 2 Using fsdb to display an i-number in idode fomat

The following command displays the i-number `386` in inode format.`386` becomes the current inode.

```
> 386:ino?i
```

例 3 Using fsdb to change the link count

The following command changes the link count for the current inode to `4`.

```
> :ln=4
```

例 4 Using fsdb to increment the link count

The following command increments the link count by 1.

```
> :ln=+1
```

例 5 Using fsdb to display the creation time as a hexadecimal long

The following command displays the creation time as a hexadecimal long.

```
> :ct=X
```

例 6 Using fsdb to display the modification time in time format

The following command displays the modification time in time format.

```
> :mt=t
```

例 7 Using fsdb to display in ASCII

The following command displays, in ASCII, block 0 of the file associated with the current inode.

```
> 0:file/c
```

例 8 Using fsdb to display the directory entries for the root inode

The following command displays the first block's directory entries for the root inode of this file system. This command stops prematurely if the EOF is reached.

```
> 2:ino,*?d
```

例 9 Using fsdb to change the current inode

The following command changes the current inode to that associated with the 5th directory entry (numbered from 0) of the current inode. The first logical block of the file is then displayed in ASCII.

```
> 5:dir:inode; 0:file,*/c
```

例 10 Using fsdb to change the i-number

The following command changes the i-number for the 7th directory slot in the root directory to 3.

```
> 2:inode; 7:dir=3
```

例 11 Using fsdb to change the name field

The following command changes the *name* field in the directory slot to name.

```
> 7:dir:nm="name"
```

例 12 Using fsdb to display the a block

The following command displays the 3rd block of the current inode as directory entries.

例 13 Using fsdb to set the contents of address

The following command sets the contents of address `2050` to `0xffffffff`. `0xffffffff` can be truncated, depending on the current type.

```
> 2050=0xffff
```

例 14 Using fsdb to place an ASCII string at an address

The following command places the ASCII string `this is some text` at address `1c92434`.

```
> 1c92434="this is some text"
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/udfs

関連項目

`clri(1M)`, `fsck_udfs(1M)`, `dir(4)`, `attributes(5)`

名前

fsdb_ufs – ufs file system debugger

形式

`fsdb -F ufs [generic_options] [specific_options] special`

機能説明

The `fsdb_ufs` command is an interactive tool that can be used to patch up a damaged UFS file system. It has conversions to translate block and i-numbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an inode. These greatly simplify the process of correcting control block entries or descending the file system tree.

`fsdb` contains several error-checking routines to verify inode and block addresses. These can be disabled if necessary by invoking `fsdb` with the `-o` option or by the use of the `o` command.

`fsdb` reads a block at a time and will therefore work with raw as well as block I/O devices. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block. Note that in order to modify any portion of the disk, `fsdb` must be invoked with the `w` option.

Wherever possible, `adb`-like syntax was adopted to promote the use of `fsdb` through familiarity.

オプション

The following option is supported:

`-o` Specify UFS file system specific options. These options can be any combination of the following separated by commas (with no intervening spaces). The options available are:

`?`	Display usage
`o`	Override some error conditions
`p='string'`	set prompt to string
`w`	open for write

使用法

Numbers are considered hexadecimal by default. However, the user has control over how data is to be displayed or accepted. The `base` command will display or set the input/output base. Once set, all input will default to this base and all output will be shown in this base. The base can be overridden temporarily for input by preceding hexadecimal numbers with '`0x`', preceding decimal numbers with '`0t`', or octal numbers with '`0`'. Hexadecimal numbers beginning with `a-f` or `A-F` must be preceded with '`0x`' to distinguish them from commands.

Disk addressing by `fsdb` is at the byte level. However, `fsdb` offers many commands to convert a desired inode, directory entry, block, superblock and so forth to a byte address. Once the address has been calculated, `fsdb` will record the result in dot (`.`).

Several global values are maintained by `fsdb`:

- the current base (referred to as `base`),

- the current address (referred to as `dot`),
- the current inode (referred to as `inode`),
- the current count (referred to as `count`),
- and the current type (referred to as `type`).

Most commands use the preset value of `dot` in their execution. For example,

```
> 2:inode
```

will first set the value of `dot` to 2, ':', will alert the start of a command, and the `inode` command will set `inode` to 2. A count is specified after a ',', Once set, `count` will remain at this value until a new command is encountered which will then reset the value back to 1 (the default). So, if

```
> 2000,400/X
```

is typed, 400 hex longs are listed from 2000, and when completed, the value of `dot` will be `2000 + 400 * sizeof (long)`. If a RETURN is then typed, the output routine will use the current values of `dot`, `count`, and `type` and display 400 more hex longs. A '*' will cause the entire block to be displayed.

End of fragment, block and file are maintained by `fsdb`. When displaying data as fragments or blocks, an error message will be displayed when the end of fragment or block is reached. When displaying data using the `db`, `ib`, `directory`, or `file` commands an error message is displayed if the end of file is reached. This is mainly needed to avoid passing the end of a directory or file and getting unknown and unwanted results.

An example showing several commands and the use of RETURN would be:

```
> 2:ino; 0:dir?d
      or
> 2:ino; 0:db:block?d
```

The two examples are synonymous for getting to the first directory entry of the root of the file system. Once there, any subsequent RETURN (or +, -) will advance to subsequent entries. Note that

```
> 2:inode; :ls
      or
> :ls /
```

is again synonymous.

Expressions

The symbols recognized by `fsdb` are:

RETURN — update the value of `dot` by the current value of `type` and display using the current value of `count`.

#	numeric expressions may be composed of +, -, *, and % operators (evaluated left to right) and may use parentheses. Once evaluated, the value of `dot` is updated.
, *count*	count indicator. The global value of `count` will be updated to `count`. The value of `count` will remain until a new command is run. A count specifier of '*' will attempt to show a *blocks's* worth of information. The default for `count` is 1.
? *f*	display in structured style with format specifier *f*. See `FormattedOutput`.
/ *f*	display in unstructured style with format specifier *f* See `FormattedOutput`.
.	the value of `dot`.
+*e*	increment the value of `dot` by the expression *e*. The amount actually incremented is dependent on the size of `type`: `dot = dot + e * sizeof (type)` The default for *e* is `1`.
-*e*	decrement the value of `dot` by the expression *e*. See +.
**e*	multiply the value of `dot` by the expression *e*. Multiplication and division don't use `type`. In the above calculation of `dot`, consider the `sizeof(type)` to be `1`.
%*e*	divide the value of `dot` by the expression *e*. See *.
< *name*	restore an address saved in register *name*. *name* must be a single letter or digit.
> *name*	save an address in register *name*. *name* must be a single letter or digit.
=*f*	display indicator. If *f* is a legitimate format specifier. then the value of `dot` is displayed using the format specifier *f*. See `FormattedOutput`. Otherwise, assignment is assumed See =.
= [*s*] [*e*]	assignment indicator. The address pointed to by `dot` has its contents changed to the value of the expression *e* or to the ASCII representation of the quoted (") string *s*. This may be useful for changing directory names or ASCII file information.
=+ *e*	incremental assignment. The address pointed to by `dot` has its contents incremented by expression *e*.
=- *e*	decremental assignment. The address pointed to by `dot` has its contents decremented by expression *e*.

Commands

A command must be prefixed by a ':' character. Only enough letters of the command to uniquely distinguish it are needed. Multiple commands may be entered on one line by separating them by a SPACE, TAB or ';'.

In order to view a potentially unmounted disk in a reasonable manner, `fsdb` offers the `cd`, `pwd`, `ls` and `find` commands. The functionality of these commands substantially matches those of its UNIX counterparts. See individual commands for details. The '*', '?', and '[-]' wild card characters are available.

`base=b`	display or set base. As stated above, all input and output is governed by the current `base`. If the `=b` is omitted, the current `base` is displayed. Otherwise, the current `base` is set to *b*. Note that this is interpreted using the old value of `base`, so to ensure correctness use the '0', '0t', or '0x' prefix when changing the `base`. The default for `base` is hexadecimal.
`block`	convert the value of `dot` to a block address.
`cd` *dir*	change the current directory to directory *dir*. The current values of `inode` and `dot` are also updated. If no *dir* is specified, then change directories to inode `2` ("/").
`cg`	convert the value of `dot` to a cylinder group.
`directory`	If the current `inode` is a directory, then the value of `dot` is converted to a directory slot offset in that directory and `dot` now points to this entry.
`file`	the value of `dot` is taken as a relative block count from the beginning of the file. The value of `dot` is updated to the first byte of this block.
`find` *dir* [`-name` *n*] [`-inum` *i*]	find files by name or i-number. `find` recursively searches directory `dir` and below for filenames whose i-number matches *i* or whose name matches pattern *n*. Note that only one of the two options (-name or -inum) may be used at one time. Also, the -print is not needed or accepted.
`fill=`*p*	fill an area of disk with pattern *p*. The area of disk is delimited by `dot` and `count`.
`fragment`	convert the value of *dot* to a fragment address. The only difference between the `fragment` command and the `block` command is the amount that is able to be displayed.
`inode`	convert the value of *dot* to an inode address. If successful, the current value of `inode` will be updated as well as the value of *dot*. As a convenient shorthand, if ':inode' appears at the beginning of the line, the value of *dot* is set to the current `inode` and that inode is displayed in inode format.
`log_chk`	run through the valid log entries without printing any information and verify the layout.

`log_delta`	count the number of deltas into the log, using the value of dot as an offset into the log. No checking is done to make sure that offset is within the head/tail offsets.
`log_head`	display the header information about the file system logging. This shows the block allocation for the log and the data structures on the disk.
`log_otodb`	return the physical disk block number, using the value of dot as an offset into the log.
`log_show`	display all deltas between the beginning of the log (BOL) and the end of the log (EOL).
`ls`	[`-R`] [`-l`] *pat1 pat2* . . . list directories or files. If no file is specified, the current directory is assumed. Either or both of the options may be used (but, if used, *must* be specified before the filename specifiers). Also, as stated above, wild card characters are available and multiple arguments may be given. The long listing shows only the i-number and the name; use the `inode` command with '?i' to get more information.
`override`	toggle the value of override. Some error conditions may be overriden if override is toggled on.
`prompt` *p*	change the `fsdb` prompt to *p*. *p* must be surrounded by (")s.
`pwd`	display the current working directory.
`quit`	quit `fsdb`.
`sb`	the value of *dot* is taken as a cylinder group number and then converted to the address of the superblock in that cylinder group. As a shorthand, ':sb' at the beginning of a line will set the value of *dot* to *the* superblock and display it in superblock format.
`shadow`	if the current inode is a shadow inode, then the value of *dot* is set to the beginning of the shadow inode data.
`!`	escape to shell

Inode Commands

In addition to the above commands, there are several commands that deal with inode fields and operate directly on the current `inode` (they still require the ':'). They may be used to more easily display or change the particular fields. The value of *dot* is only used by the '`:db`' and '`:ib`' commands. Upon completion of the command, the value of *dot* is changed to point to that particular field. For example,

```
> :ln=+1
```

would increment the link count of the current `inode` and set the value of *dot* to the address of the link count field.

`at` access time.

`bs` block size.

`ct` creation time.

`db` use the current value of *dot* as a direct block index, where direct blocks number from 0 - 11. In order to display the block itself, you need to 'pipe' this result into the `block` or `fragment` command. For example,

```
> 1:db:block,20/X
```

would get the contents of data block field 1 from the inode and convert it to a block address. 20 longs are then displayed in hexadecimal. See `FormattedOutput`.

`gid` group id.

`ib` use the current value of *dot* as an indirect block index where indirect blocks number from 0 - 2. This will only get the indirect block itself (the block containing the pointers to the actual blocks). Use the `file` command and start at block 12 to get to the actual blocks.

`ln` link count.

`mt` modification time.

`md` mode.

`maj` major device number.

`min` minor device number.

`nm` although listed here, this command actually operates on the directory name field. Once poised at the desired directory entry (using the *directory* command), this command will allow you to change or display the directory name. For example,

```
> 7:dir:nm="foo"
```

will get the 7th directory entry of the current `inode` and change its name to foo. Note that names cannot be made larger than the field is set up for. If an attempt is made, the string is truncated to fit and a warning message to this effect is displayed.

`si` shadow inode.

`sz` file size.

`uid` user id.

Formatted Output

There are two styles and many format types. The two styles are structured and unstructured. Structured output is used to display inodes, directories, superblocks and the like. Unstructured displays raw data. The following shows the different ways of displaying:

`?`

- `c` display as cylinder groups
- `i` display as inodes
- `d` display as directories
- `s` display as superblocks
- `S` display as shadow inode data

`/`

- `b` display as bytes
- `c` display as characters
- `o O` display as octal shorts or longs
- `d D` display as decimal shorts or longs
- `x X` display as hexadecimal shorts or longs

The format specifier immediately follows the '/' or '?' character. The values displayed by '/b' and all '?' formats are displayed in the current `base`. Also, `type` is appropriately updated upon completion.

使用例

例 1 Displaying in Decimal

The following command displays `2010` in decimal (use of `fsdb` as a calculator for complex arithmetic):

```
> 2000+400%(20+20)=D
```

例 2 Displaying an i-number in Inode Format

The following command displays i-number `386` in an inode format. This now becomes the current `inode`:

```
> 386:ino?i
```

例 3 Changing the Link Count

The following command changes the link count for the current `inode` to `4`:

```
> :ln=4
```

例 4 Incrementing the Link Count

The following command increments the link count by 1:

```
> :ln=+1
```

例 5 Displaying the Creation Time

The following command displays the creation time as a hexadecimal long:

```
> :ct=X
```

例 6 Displaying the Modification Time

The following command displays the modification time in time format:

```
> :mt=t
```

例 7 Displaying in ASCII

The following command displays in ASCII, block zero of the file associated with the current inode:

```
> 0:file/c
```

例 8 Displaying the First Block's Worth of Directorty Entries

The following command displays the first block's worth of directory entries for the root inode of this file system. It will stop prematurely if the EOF is reached:

```
> 2:ino,*?d
```

例 9 Displaying Changes to the Current Inode

The following command displays changes the current inode to that associated with the 5th directory entry (numbered from zero) of the current inode. The first logical block of the file is then displayed in ASCII:

```
> 5:dir:inode; 0:file,*/c
```

例 10 Displaying the Superblock

The following command displays the superblock of this file system:

```
> :sb
```

例 11 Displaying the Cylinder Group

The following command displays cylinder group information and summary for cylinder group 1:

```
> 1:cg?c
```

例 12 Changing the i-number

The following command changes the i-number for the seventh directory slot in the root directory to 3:

```
> 2:inode; 7:dir=3
```

例 13 Displaying as Directory Entries

The following command displays the third block of the current `inode` as directory entries:

```
> 2:db:block,*?d
```

例 14 Changing the Name Field

The following command changes the name field in the directory slot to *name*:

```
> 7:dir:nm="name"
```

例 15 Getting and Filling Elements

The following command gets fragment `3c3` and fill `20` `type` elements with `0x20`:

```
> 3c3:fragment,20:fill=0x20
```

例 16 Setting the Contents of an Address

The following command sets the contents of address `2050` to `0xffffffff`. `0xffffffff` may be truncated depending on the current `type`:

```
> 2050=0xffff
```

例 17 Placing ASCII

The following command places the ASCII for the string at `1c92434`:

```
> 1c92434="this is some text"
```

例 18 Displaying Shadow Inode Data

The following command displays all of the shadow inode data in the shadow inode associated with the root inode of this file system:

```
> 2:ino:si:ino;0:shadow,*?S
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

clri(1M), fsck_ufs(1M), dir_ufs(4), attributes(5), ufs(7FS)

警告

Since fsdb reads the disk raw, extreme caution is advised in determining its availability of fsdb on the system. Suggested permissions are 600 and owned by bin.

注意事項

The old command line syntax for clearing i-nodes using the ufs-specific '-z i-number' option is still supported by the new debugger, though it is obsolete and will be removed in a future release. Use of this flag will result in correct operation, but an error message will be printed warning of the impending obsolesence of this option to the command. The equivalent functionality is available using the more flexible clri(1M) command.

名前

fsirand – install random inode generation numbers

形式

`fsirand [-p]` *special*

機能説明

`fsirand` installs random inode generation numbers on all the inodes on device *special*, and also installs a file system ID in the superblock. This helps increase the security of file systems exported by NFS.

`fsirand` must be used only on an unmounted file system that has been checked with `fsck(1M)` The only exception is that it can be used on the root file system in single-user mode, if the system is immediately re-booted afterwards.

オプション

`-p` Print out the generation numbers for all the inodes, but do not change the generation numbers.

使用法

See `largefile(5)` for the description of the behavior of `fsirand` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fsck(1M)`, `attributes(5)`, `largefile(5)`

名前

fssnap – create temporary snapshots of a file system

形式

```
fssnap [-F FSType] [-V] -o special_options /mount/point

fssnap  -d [-F  FSType] [-V] /mount/point | dev

fssnap -i [-F FSType] [-V] [-o special_options]
      [/mount/point | dev]
```

機能説明

The `fssnap` command creates a stable, read-only snapshot of a file system when given either an active mount point or a special device containing a mounted file system, as in the first form of the synopsis. A snapshot is a temporary image of a file system intended for backup operations.

While the snapshot file system is stable and consistent, an application updating files when the snapshot is created might leave these files in an internally inconsistent, truncated, or otherwise unusable state. In such a case, the snapshot will contain these partially written or corrupted files. It is a good idea to ensure active applications are suspended or checkpointed and their associated files are also consistent during snapshot creation.

File access times are not updated while the snapshot is being created.

A path to the virtual device that contains this snapshot is printed to standard output when a snapshot is created.

オプション

The following options are supported:

`-d`	Deletes the snapshot associated with the given file system.
`-F` *FSType*	Specifies the file system type to be used. The `FSType` should either be specified here or be determined by matching the block special device with an entry in the `/etc/vfstab` table, or by consulting `/etc/default/fs`.
`-i`	Displays the state of any given `FSType` snapshot. If a mount-point or device is not given, a list of all snapshots on the system is displayed. When a mount-point or device is specified, detailed information is provided for the specified file system snapshot by default. The format and meaning of this information is file-system dependent. See the `FSType`-specific `fssnap` man page for details.
`-o` *special_options*	See the `FSType`-specific man page for `fssnap`.
`-V`	Echoes the complete command line, but does not execute the command.

オペランド

The following operands are supported:

/mount/point	The directory where the file system resides.

使用例

See FSType-specific man pages for examples.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

ファイル

/etc/vfstab Specifies file system type.

/etc/default/fs Specifies the default local file system type.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

fssnap_ufs(1M), attributes(5)

注意事項

This command might not be supported for all FSTypes.

名前

fssnap_ufs – create a temporary snapshot of a UFS file system

形式

```
fssnap [-F ufs] [-V] -o backing-store=path,
      [specific-options] /mount/point

fssnap -d [-F ufs] [-V] /mount/point | dev

fssnap -i [-F ufs] [-V] [-o specific-options] /mount/point | dev
```

機能説明

The `fssnap` command queries, creates, or deletes a temporary snapshot of a UFS file system. A snapshot is a point-in-time image of a file system that provides a stable and unchanging device interface for backups.

When creating a file system snapshot, you must specify the file system to be captured and the backing-store file. The backing-store file(s) are where the snapshot subsystem saves old file system data before it is overwritten. Beyond the first backing-store file, `fssnap` automatically creates additional backing-store files on an as-needed basis.

The number and size of the backing store files varies with the amount of activity in the file system. The destination path must have enough free space to hold the backing-store file(s). This location must be different from the file system that is being captured in a snapshot. The backing-store file(s) can reside on any type of file system, including another UFS file system or an NFS–mounted file system.

オプション

The following options are supported:

`-d`
: Deletes the snapshot associated with the given file system.

`-i`
: Displays the state of one or all UFS snapshots. If a mount-point or device is not specified, a list of all snapshots on the system is displayed. When a mount-point or device is specified, detailed information is provided for the specified file system snapshot by default.

  Use the `-o` options with the `-i` option to specify what snapshot information is displayed. Since this feature is provided primarily for use in scripts and on the command line, no labels are displayed for the data. Sizes are all in bytes, and the output is not internationalized or localized. The information is displayed on one line per option. Unrecognized options display a single ? on the line. One line per option guarantees that there are the same number of lines as options specified and there is a one–to-one correspondence between an output line and an option.

  The following `-o` options display specific information for a given snapshot. See the EXAMPLES section for examples of how to use these options.

  `snapnumber`
  : Display the snapshot number.

  `blockdevname`
  : Display the block device path.

`rawdevname`
: Display the raw device path.

`mountpoint`
: Display the mount point of the master file system.

`state`
: Display the state of the snapshot device.

`backing-store`
: Display the location of the first backing-store file for this snapshot. If there are multiple backing-store files, subsequent files have the same name as the first file, with the suffixes `.2`, `.3`, and so forth.

`backing-store-len`
: Display the sum of the sizes of the backing-store files.

`maxsize`
: Display the `maxsize` value specified for the backing-store file(s).

`createtime`
: Display the time that the snapshot was created.

`chunksize`
: Display the copy-on-write granularity.

`-o` *specific-options*
: Without `-d` or `-i`, the default action is to create a snapshot. Specify the following options when creating a snapshot. All of these options are discretionary, except for the backing-store file, which is required.

`backing-store=`*path*
: Uses *path* in the creation of the backing-store file(s). *path* must not reside on the file system that is being captured in a snapshot and must not be the name of an existing file. If *path* is a directory, then a backing-store file is created within it using a name that is generated automatically. If *path* is not a directory and does not already exist, then a backing-store file with that name is created. If more than one backing-store file is required, `fssnap` creates subsequent files automatically. The second and subsequent files have the same name as the first file, with suffixes of `.2`, `.3`, and so forth.

  This option can be abbreviated as `bf=`*path* or `bs=`*path*.

`unlink`
: Unlinks the backing-store file after the snapshot is created. This option specifies that the backing-store file does not need to be removed manually when the snapshot is deleted. This might make administration more difficult since the file is not visible in the file system. If this option is not specified, the backing-store files should be removed manually after the snapshot is deleted.

`chunksize=`*n* [`k`,`m`,`g`]
: Uses *n* for the chunk size. Chunk size is the granularity of the data that is sent to the backing store.

  Specify `chunksize` in the following units: `k` for kilobytes, `m` for megabytes, or `g` for gigabytes. By default, chunk size is four times the block size of the file system (typically `32k`).

`maxsize=`*n*[`k`,`m`,`g`]
: Does not allow the sum of the sizes of the backing-store file(s) to exceed *n*, where *n* is the unit specified. The snapshot is deleted automatically when the sum of the sizes of the backing-store file(s) exceeds `maxsize`.

  Specify `maxsize` in the following units: `k` for kilobytes, `m` for megabytes, or `g` for gigabytes.

`raw`
: Displays to standard output the name of the raw device instead of the block device when a snapshot is created. The block device is printed by default (when `raw` is not specified). This option makes it easier to embed `fssnap` commands in the command line for commands that require the raw device instead. Both devices are always created. This option affects only the output.

オペランド

The following operands are supported:

mount-point
: The directory where the file system resides.

special
: The physical device for the file system, such as `/dev/dsk/c0t0d0s7`.

使用例

例 1　Creating a Snapshot of a File System

The following example creates a snapshot of a file system. The block special device created for the snapshot is `/dev/fssnap/0`.

```
# fssnap -F ufs -o backing-store=/var/tmp /export/home
/dev/fssnap/0
```

例 2　Backing Up a File System Snapshot Without Having To Unmount the File System

The following example backs up a file system snapshot without having to unmount the file system. Since `ufsdump` requires the path to a raw device, the `raw` option is used. The `/export/home` file system snapshot is removed in the second command.

```
# ufsdump 0uf /dev/rmt/0 ‘fssnap -F ufs
      -o raw,bs=/export/snap /export/home‘
<output from ufsdump>
# fssnap -F ufs -d /export/home
```

例 3 Backing Up a File System

When backing up a file system, do not let the backing-store file(s) exceed 400 Mbytes. The second command removes the /export/home file system snapshot.

```
# ufsdump 0uf /dev/rmt/0 ‘fssnap -F ufs
      -o maxsize=400m,backing-store=/export/snap,raw
      /export/home‘
# fssnap -F ufs -d /export/home
```

例 4 Performing an Incremental Dump of a Snapshot

The following example uses ufsdump to back up a snapshot of /var. Note the use of the N option to ufsdump, which writes the name of the device being dumped, rather than the name of the snapshot device, to /etc/dumpdates file. See ufsdump(1M) for details on the N flag.

```
# ufsdump lfNu /dev/rmt/0 /dev/rdsk/c0t3d0s2 ‘fssnap -F ufs
-o raw,bs=/export/scratch,unlink /var‘
```

例 5 Finding Out What Snapshots Currently Exist

The following command displays the currently existing snapshots.

```
# fssnap -i
0  /src
1  /export/home
<output continues>
```

例 6 Mounting a File System Snapshot

The following example creates a file system snapshot. After you create a file system snapshot, mount it on /tmp/mount for temporary read-only access.

```
# fssnap -F ufs -o backing-store=/nfs/server/scratch /export/home
/dev/fssnap/1
# mkdir /tmp/mount
# mount -F ufs -o ro /dev/fssnap/1 /tmp/mount
```

例 7 Creating a File System Snapshot and Unlinking the Backing-store File

The following example creates a file system snapshot and unlinks the backing-store file. After creating a file system snapshot and unlinking the backing-store file, check the state of the snapshot.

```
# fssnap -o bs=/scratch,unlink /src
/dev/fssnap/0
# fssnap -i /src
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /src
Device state                  : active
```

例 7 Creating a File System Snapshot and Unlinking the Backing-store File (続き)

```
Backing store path            : /scratch/snapshot2 <UNLINKED>
Backing store size            : 192 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Sat May 06 10:55:11 2000
Copy-on-write granularity     : 32 KB
```

例 8 Displaying the Size and Location of the Backing-store File(s) and the Creation Time for the Snapshot

The following example displays the size of the backing-store file(s) in bytes, the location of the backing store, and the creation time for the snapshot of the `/test` file system.

```
# fssnap -i -o backing-store-len,backing-store,createtime /test
196608
/snapshot2
Sat May 6 10:55:11 2000
```

Note that if there are multiple backing-store files stored in `/snapshot2`, they will have names of the form *file* (for the first file), *file*.1, *file*.2, and so forth.

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

The script-readable output mode is a stable interface that can be added to, but will not change. All other interfaces are subject to change.

関連項目

`mlock(3C)`, `attributes(5)`

See the `ntpd` man page, delivered in the `SUNWntpu` package (not a SunOS man page).

注意事項

The `fssnap` device files should be treated like a regular disk block or character device.

The association between a file system and the snapshot is lost when the snapshot is deleted or the system reboots. Snapshot persistence across reboots is not currently supported.

To avoid unnecessary performance impacts, perform the snapshot and system backup when the system is least active.

It is not possible to perform a snapshot of a file system if any of the following conditions are true:

- The file system is in use by system accounting
- The file system contains a local swap file
- The file system is used as backing store by an application that uses `mlock(3C)` to lock its pages. Typically, these are real time applications, such as `ntpd` (delivered in the `SUNWntpu` package).

These conditions result in `fssnap` being unable to write lock the file system prior to performing the snapshot.

名前

fsstat – report file system statistics

形式

```
fsstat [-a|f|i|n|v] [-T | u|d] {-F | {fstype|path}...}
      [interval [count]]
```

機能説明

`fsstat` reports kernel file operation activity by the file system type (*fstype*) or by the path name, which is converted to a mount point. The first set of lines of output reports all activity since:

- The file system module was loaded (in the case of *fstype*)
- The file system was mounted (in the case of mount point)

Statistics are gathered at the file system independent layer at both the *fstype* and the mount point levels. However, not all file system types are represented in the gathering of statistics. (See the NOTES section of this man page.)

The output of `fsstat` is dependent on the mode (option) requested. All statistic fields are displayed using "smart numbers" which automatically scale the units in a human readable form that fits in a maximum of 5 characters. For example:

100	is displayed as 100
2048	is displayed as 2K
3000000	is displayed as 2.86M

The unit modifiers are: K (Kbyte), M (Mbyte), G (Gbyte), T (terabyte), P (petabyte), and E (exabyte).

During the execution of `fsstat`, the state of the system can change. If relevant, a state change message is included in the `fsstat` output in one of the following forms:

```
<<mount point no longer available: {path}>>
<<file system module no longer loaded: {fstype}>>
```

After the state change messages are displayed, `fsstat` continues to display the statistics as directed. If all of the *fstypes* and mount points that `fsstat` was reporting on are no longer available, then `fsstat` exits.

The user is required to specify the `-F` option (all available file system types) or a list of one or more *fstypes* and/or mount points.

The default report shows general file system activity. This display combines similar operations into general categories as follows:

new file	Number of creation operations for file system objects (for example, files, directories, symlinks, etc.)
name remov	Number of name removal operations
name chng	Number of name change operations

attr get	Number of object attribute retrieval operations
attr set	Number of object attribute change operations
lookup ops	Number of object lookup operations
rddir ops	Number of read directory operations
read ops	Number of data read operations
read bytes	Bytes transferred by data read operations
write ops	Number of data write operations
write bytes	Bytes transferred by data write operations

The entity being reported on (*fstype* or mount point) is displayed in the last column.

オプション

The following options are supported:

`-a` Report the activity for kernel attribute operations. The following statistics are reported:

getattr	Number of file attribute retrieval calls
setattr	Number of file attribute modification calls
getsec	Number of file security attribute retrieval calls
setsec	Number of file security attribute modification calls

The entity being reported on (*fstype* or mount point) is displayed in the last column.

`-f` Report the full activity for all kernel file operations. Each file operation is listed in the left column. The following statistics are reported for each operation:

#ops	Number of calls for this operation
bytes	Average transfer size in bytes (only applies to read, write, readdir)

The entity being reported on (*fstype* or mount point) is displayed in the first row.

`-i` Reports the activity for kernel I/O operations. The following statistics are reported:

read ops	Number of data read calls
read bytes	Number of bytes read
write ops	Number of data write calls
write bytes	Number of bytes written
rddir ops	Number of read directory calls

rddir bytes	Number of bytes read by reading directories
rwlock ops	Number of internal file system lock operations
rwulock ops	Number of internal file system unlock operations

The entity being reported on (*fstype* or mount point) is displayed in the last column.

-n Reports the activity for kernel naming operations. The following statistics are reported:

lookup	Number of file name retrieval calls
creat	Number of file creation calls
remov	Number of file remove calls
link	Number of link calls
renam	Number of file renaming calls
mkdir	Number of directory creation calls
rmdir	Number of directory removal calls
rddir	Number of directory read calls
symlink	Number of symlink creation calls
rdlink	Number of symlink read calls

The entity being reported on (*fstype* or mount point) is displayed in the last column.

-v Reports the activity for calls to the virtual memory operations. The following statistics are reported.

map	Number of calls mapping a file
addmap	Number of calls setting additional mapping to a mapped file
delmap	Number of calls deleting mapping to a file
getpag	Number of calls retrieving a page of data from a file
putpag	Number of calls writing a page of data to a file
pagio	Number of calls to transfer pages in file system swap files

The entity being reported on (fstype or mount point) is displayed in the last column.

-F Report on all available file system types.

-T *u*|*d* Display a time stamp.

Specify *u* for a printed representation of the internal representation of time (see `time(2)`) Specify *d* for the standard date format. (See `date(1)`). The time stamp is only used when an interval is set.

オペランド

The following operands are supported:

count Display only *count* reports.

fstype Explicitly specify the file system type(s) to be reported. The file system module must be loaded.

interval Report once each *interval* seconds.

path Specify the path(s) of the mount point(s) to be reported. If path is not a mount point, the mount point containing path will be determined and displayed in the output.

If no *interval* and no *count* are specified, a single report is printed and `fsstat` exits. If an *interval* is specified but no *count* is specified, `fsstat` prints reports every *interval* seconds indefinitely until the command is interrupted.

使用例

例 1 Displaying General Activity

The following example shows general activity for all file system types.

```
$ fsstat -F
 new  name   name  attr   attr lookup rddir  read read  write write
 file remov  chng   get    set    ops   ops   ops bytes   ops bytes
  313K  214K 38.5K 2.16M 56.2K  8.36M 52.8K 19.7M 39.9G 18.8M 39.1G ufs
     0     0     0 2.95K     0  3.81K   282 2.52K  466K     0     0 proc
     0     0     0     0     0      0     0     0     0     0     0 nfs
    10     8     2    86     9     98    15   413  103M 8.43K 1.05G zfs
    13    14     4    98    16    125    10 1.01K  258M 15.9K  127M lofs
8.73K 3.29K 5.25K 55.3K    37  1.20M    44 37.9K 38.3M 47.2K 35.9M tmpfs
     0     0     0 4.93K     0      0     0 1.08K  913K     0     0 mntfs
     3     2     1   503     3    897    13   122 25.8K   128  272K nfs3
    10     8     0   615    10  10.1K    18    61 45.6K   292 2.26M nfs4
```

例 2 Displaying Naming Activity

The following example shows the naming activity for ufs, nfs, nfs3, nfs4, and tmpfs:

```
$ fsstat -n ufs nfs nfs3 nfs4 tmpfs
lookup creat remov  link renam mkdir rmdir rddir symlnk rdlnk
3.57M  3.10K   586     6    24   115   100 30.2K      5  330K ufs
    0      0     0     0     0     0     0     0      0     0 nfs
18.3K      3     5     0     0     0     0 1.03K      2   346 nfs3
  535      0     0     0     0     0     0    46      0     4 nfs4
```

例 2 Displaying Naming Activity *(続き)*

```
  146     24    15     0     0     4     0     4     0     0 tmpfs
```

例 3 Displaying Attribute Activity

The following example shows the attribute activity for the FS type ufs and the mounted file systems "/" and "/export/home" every three seconds for every third iteration:

```
# fsstat -a ufs / /export/home 3 3
getattr setattr getsec setsec
  378K    91.9K  11.8K      0 ufs
  367K    82.3K  11.6K      0 /
 11.3K     9.6K    198      0 /export/home
 4.97K    2.27K    163      0 ufs
 3.94K    1.36K    162      0 /
 1.03K      927      1      0 /export/home
 2.30K    1.06K     73      0 ufs
 1.95K      766     71      0 /
   361      317      2      0 /export/home
 2.33K    1.06K     78      0 ufs
 1.64K      451     77      0 /
   711      631      1      0 /export/home
```

例 4 Displaying File Operation Statistics

The following example shows the statistics for each file operation for "/" (using the `-f` option):

```
$ fsstat -f /
Mountpoint: /
 operation  #ops  bytes
      open 8.54K
     close  9.8K
      read 43.6K  65.9M
     write 1.57K  2.99M
     ioctl 2.06K
     setfl     4
   getattr 40.3K
   setattr    38
    access 9.19K
    lookup  203K
    create   595
    remove    56
      link     0
    rename     9
     mkdir    19
     rmdir     0
   readdir 2.02K  2.27M
   symlink     4
```

例 4 Displaying File Operation Statistics (続き)

```
  readlink 8.31K
     fsync   199
  inactive 2.96K
       fid     0
    rwlock 47.2K
  rwunlock 47.2K
      seek 29.1K
       cmp 42.9K
    frlock 4.45K
     space     8
    realvp 3.25K
   getpage  104K
   putpage 2.69K
       map 13.2K
    addmap 34.4K
    delmap 33.4K
      poll   287
      dump     0
  pathconf    54
    pageio     0
   dumpctl     0
   dispose 23.8K
getsecattr   697
setsecattr     0
   shrlock     0
   vnevent     0
```

環境変数

See environ(5) for descriptions of the following environment variables that affect the execution of fsstat: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, LC_TIME, and NLSPATH.

終了ステータス

The following exit values are returned:

0 Successful completion.

1 A fatal error occurred. A fatal error could be a failed system call or another internal error.

2 Invalid command-line options were specified.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	See below.

The command-line options are Uncommitted. The human-readable output is not considered an interface.

関連項目

date(1), time(2), attributes(5)

注意事項

All display options (-a, -f, -i, -n, -v) are mutually exclusive. Entering more than one of these options will result in an error.

The *fstype* and *path* operands must appear after the option, but before the *interval* or *count* on the command line. For example, "fsstat -a *fstype interval*". Preference is given to *fstype* so that if a user wishes to see the statistics for a directory that has the same name as an *fstype* (for example, ufs), then the path must be specified unambiguously (for example, ./ufs). Similarly, in order to define a file with a numeric name (for example, "10") from an interval or count operand, the name should be prefixed accordingly (for example, ./10).

When an interval is used, headers repeat after more than 12 lines of statistics have been displayed and the set of lines to be displayed in the current interval have completed.

Statistics are not displayed for all pseudo-filesystems. The output displayed with the -F option shows which of the loaded filesystem types are supported.

Unbundled file systems may not be recognized by fsstat.

The command-line options are classified as Uncommitted and could change. The output is not considered to be an interface. The construction of higher level software tools depend on either the command-line options or the output of fsstat is not recommended.

名前

fstyp – determine file system type

形式

`fstyp [-a | -v]` *special* `[:`*logical-drive*`]`

機能説明

`fstyp` allows the user to determine the file system type of unmounted file systems using heuristic programs.

An `fstyp` module for each file system type to be checked is executed; each of these modules applies an appropriate heuristic to determine whether the supplied *special* file is of the type for which it checks. If it is, the program prints on standard output the usual file system identifier for that type (for example, “ufs") and exits with a return code of `0`; if none of the modules succeed, the error message `unknown_fstyp (no matches)` is returned and the exit status is `1`. If more than one module succeeds, the error message `unknown_fstyp (multiple matches)` is returned and the exit status is `2`. Other errors are printed in the same format.

This command is unreliable and its results should not be used to make any decisions about subsequent use of a storage device or disk partition.

オプション

`-a` Output all available file system attributes. If a file system has been successfully identified, and this option is specified, the *FSType* identifier is followed by one or more “name-value” pairs, one per line, in the format:

```
name: value
```

The following conventions are recognized for the file system attributes:

- String values are put in single quotes.
- Nested “name-value” list increases the indentation of its values by four whitespaces.
- For an array of “name-value” pairs, one array entry is printed per line, with the index following the name in square brackets.

For instance, in the following example, “*top_list*” is a “name-value” list, consisting of a string array “*string_array*” and a “name-value” list array “*list_array*”. The second “list_array”element is an integer array “*int_array*” containing three elements.

```
top_string: 'String'
top_list:
    string_array[0]: 'S0'
    string_array[1]: 'S1'
    list_array[0]:
        int_one: 1
        string_two: 'S2'
    list_array[1]:
        int_array[0]: 1
        int_array[1]: 2
        int_array[2]: 3
```

In addition to the *FSType*-specific attributes, the following generic attributes may be present:

gen_clean Allowable values are “true” or “false”. If the value is “false”, the file system is damaged or was not unmounted cleanly and the `fsck(1M)` command must be run before this file system can be mounted.

gen_guid Globally unique identifier. This string uniquely identifies the file system.

gen_version A string describing the file system version.

gen_volume_label Volume label, a human-readable string used to either describe or identify the file system.

`-v` Produce verbose output. This is usually information about the file systems superblock and varies across different *FSTypes*. See `ufs(7FS)`, `mkfs_ufs(1M)`, and `tunefs(1M)` for details.

使用法

See `largefile(5)` for the description of the behavior of `fstyp` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fsck(1M)`, `mkfs_ufs(1M)`, `tunefs(1M)`, `attributes(5)`, `libfstyp(3LIB)`, `largefile(5)`, `hsfs(7FS)`, `ufs(7FS)`, `pcfs(7FS)`

注意事項

The use of heuristics implies that the result of `fstyp` is not guaranteed to be accurate.

This command is unreliable and its results should not be used to make any decisions about subsequent use of a storage device or disk partition.

名前

fuser – ファイルまたはファイル構造を使用しているプロセスの特定

形式

```
/usr/sbin/fuser [-c | -d | -f] [-nu] [-k | -s sig] files
     [ [- ] [-c | -d | -f] [-nu] [-k | -s sig] files] ...
```

機能説明

fuser は、引数として指定された *files* を使用しているプロセスのプロセス ID を表示します。

各プロセス ID には文字修飾詞が続きます。文字修飾詞には以下のような意味があり、プロセスがファイルをどのように使用しているかを示します。

- c　カレントディレクトリとしてファイルを使用している
- m　mmap(2) によってマップされたファイルを使用している。詳細は mmap(2) を参照
- n　そのファイルに対して、非ブロッキング強制ロックを保持している
- o　オープンファイルとしてファイルを使用している
- r　ルートディレクトリとしてファイルを使用している
- t　テキストファイルとしてファイルを使用している
- y　制御端末としてファイルを使用している

ファイルシステムがマウントされているブロック型特殊デバイスについては、そのデバイス上のファイルを使用しているすべてのプロセスが表示されます。その他のタイプのファイル (テキストファイル、実行可能ファイル、ディレクトリ、デバイスなど) については、そのファイルを使用しているプロセスだけが報告されます。

fuser は、すべてのタイプのデバイスに対して、デバイスを開いている既知のカーネル消費者も表示します。カーネル消費者は、次のうちの 1 つの形式で表示されます。

[*module_name*]
[*module_name*,dev_path=*path*]
[*module_name*,dev=(*major*,*minor*)]
[*module_name*,dev=(*major*,*minor*),dev_path=*path*]

複数のファイルグループを指定する場合は、ファイルグループごとにオプションを指定できます。現在使用中のオプションを取り消す場合は、単独のダッシュを使用します。

プロセス ID は、スペースで区切られた単一行として標準出力に出力されます。末尾の単一の改行など、その他の出力はすべて標準エラーに書き込まれます。

fuser はだれでも実行できますが、他のユーザーのプロセスを終了させることができるのは、スーパーユーザーだけです。

オプション

次のオプションがサポートされています。

- `-c` ファイルシステムのマウントポイントであるファイル、およびマウントされているファイルシステム内のすべてのファイルについて報告します。
- `-d` 指定したマイナーノードと同じデバイスノードに関連するすべてのマイナーノードについて、デバイス使用状況を報告します。このオプションは、マウントされているファイルシステム内にあるファイルについてのファイル使用状況は報告しません。
- `-f` マウントされているファイルシステム内のファイルではなく、指定したファイルについて報告します。
- `-k` 各プロセスに `SIGKILL` シグナルを送ります。このオプションは各プロセスに kill コマンドを発行するので、強制終了メッセージがすぐに表示されないことがあります (`kill(2)` を参照)。カーネルファイル消費者にはシグナルは送られません。
- `-n` ファイルに対して非ブロッキング強制ロックを保持しているプロセスだけを表示します。
- `-s` *sig* 各プロセスにシグナルを送ります。*sig* オプション引数には、`<signal.h>` ヘッダーで定義されている記号名の 1 つを指定するか、または 10 進整数のシグナル番号を指定します。*sig* が記号名で、`SIG` 接頭辞を付けずに、大文字小文字の区別なしで認識されます。`-k` オプションは `-s KILL` または `-s 9` と同等です。カーネルファイル消費者にはシグナルは送られません。
- `-u` プロセス ID の後に、括弧で囲んだユーザーのログイン名を表示します。

使用例

例 1 マウントポイントおよびファイルについて報告する

次の例は、マウントポイントおよびマウントされているファイルシステム内のファイルについて報告します。

```
example% fuser -c /export/foo
```

例 2 マウントポイントおよびファイルについて報告するときの出力を制限する

次の例は、マウントポイントおよびマウントされているファイルシステム内のファイルについて報告し、その出力を非ブロッキング強制ロックを保持しているプロセスに限定します。

```
example% fuser -cn /export/foo
```

例 3 非ブロッキング強制ロックを保持しているプロセスへ SIGTERM を送信する

次のコマンドは、ファイル `/export/foo/my_file` に対して非ブロッキング強制ロックを保持しているすべてのプロセスに、`SIGTERM` を送ります。

```
example% fuser -fn -s term /export/foo/my_file
```

環境変数

fuser の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。LANG、LC_ALL LC_CTYPE、LC_MESSAGES、および NLSPATH。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実
標準	standards(5) を参照してください。

関連項目

ps(1), mount(1M), kill(2), mmap(2), signal(3C), attributes(5), environ(5), standards(5)

注意事項

fuser はシステムイメージのスナップショットで動作するため、fuser の実行中にファイルを使用し始めたプロセスを見逃してしまうことがあります。また、ファイルを使用していると報告されたプロセスが、fuser の実行中にファイルの使用を中止していることもあります。このような理由から、-k オプションの使用は勧められません。

名前

fwflash – firmware query and update utility

形式

```
/usr/sbin/fwflash  [-l [-c device_class | ALL ]]
     | [-v] | [-h]

fwflash  [-f file1,file2,file3,... | -r file]
     [-y] [-d device_path]
```

機能説明

The `fwflash` command writes a binary image file to supported flashable devices attached to a Solaris host. It also provides the ability to read firmware to a file if supported by the device. Because changing the firmware in a device can have significant impact on the stability of a system, only users with the privilege `All` are allowed to execute this command. Users authorized to run `fwflash` can be granted the "Firmware Flash Update" Rights Profile.

The first form of the command, above, provides information about devices. It lists all devices currently available on the system that are supported by `fwflash` for firmware upgrade. You can filter the list operation, to display only specified classes of devices. The second form of the command provides the operations to read or write the firmware images to specific devices.

オプション

The following options are supported:

`-c` *device_class*
: An optional parameter, valid only when used with the `-l` option. This option causes the command to list only devices of a specific class type. Currently supported classes are `IB`, `ses`, `sesgen`, or `ALL`. If `-c` is not specified for the `-l`option, the class defaults to `ALL`.

`-d` *dev_path*
: The *dev_path* is absolute path name of the device that the user wants to modify with the `-f`or `-r` operation. If the device cannot be found, the command fails. If the `-d` option is specified, then either `-f` or `-r` must also be specified.

`-f` *file1*,*file2*,*file3*,...
: Specify the path to one or more binary firmware image files you want to write to the device. `fwflash` will verify that each file is a valid firmware image for the specified device. If it is not, the command fails with an appropriate error message.

  If multiple firmware image files are specified, each image is verified and flashed to the device in the order given on the command line. If any of the specified files cannot be successfully flashed, then an appropriate message is displayed.

  After a new firmware image is flashed to a device, a reboot is required to correctly activate the new firmware.

`-h`
: Display the command line usage message for `fwflash`.

`-l`
: List the devices on a system available for firmware upgrade and display information specific to each device or device class.

For InfiniBand (IB) devices, the list operation displays the guids (Globally Unique Identifier) currently set for the HCA, as well as the current firmware revision installed. There are four separate guids on the HCA; two of them can be set with the same value.

For SCSI Enclosure Services (ses or sgen) devices, an identifying target-port worldwide name is displayed, if available.

-r *file*
Specify the path to a file to create when reading the firmware from the device. The -f and -r options are mutually exclusive.

Not all flashable devices support reading firmware images back from the device. At present, only InfiniBand (IB) devices are supported for this operation. A message will be displayed if the selected device does not support this operation.

-v
Display fwflash version information and exit.

-y
Valid only when a flash read (-r) or write (-f) operation is specified. This option causes fwflash not to prompt for confirmation during operation and operate non-interactively. Note that there is no option that allows you to forcibly flash an incompatible firmware image onto a device.

使用例

例 1 Entering Command Without Arguments

The following command shows fwflash when the command is entered without arguments.

```
example# fwflash
Usage:
Usage:
      fwflash [-l [-c device_class | ALL]] | [-v] | [-h]
      fwflash [-f file1,file2,file3,... | -r file] [-y] -d device_path

      -l              list flashable devices in this system
      -c device_class limit search to a specific class
                      eg IB for InfiniBand, ses for SCSI Enclosures
      -v              print version number of fwflash utility
      -h              print this usage message

      -f file1,file2,file3,...
                      firmware image file list to flash
      -r file         file to dump device firmware to
      -y              answer Yes/Y/y to prompts
      -d device_path  pathname of device to be flashed

      If -d device_path is specified, then one of -f <files>
      or -r <file> must also be specified
```

例 1 Entering Command Without Arguments *(続き)*

```
If multiple firmware images are required to be flashed
they must be listed together, separated by commas. The
images will be flashed in the order specified.
```

例 2 Listing Devices Available to Flash

The following command lists the devices available to be flashed.

```
example# fwflash -l
List of available devices:
Device[0],  /devices/pci@0,0/pci8086,3595@2/pci8086,32a@0,2/\
            pci15b3,5a46@c/pci15b3,5a44@0:devctl
    Class [IB]
        GUID: System Image - 0002c901081e33b3
              Node         - 0000000000003446
              Port 1       - 0002c901081e33b1
              Port 2       - 0002c901081e33b2
        Firmware revision: 3.3.0002
        No hardware information available for this device

Device[1],  /devices/pci@0,0/pci8086,3597@4/pci15b3,6278@0:devctl
    Class [IB]
        GUID: System Image - 0002c9010a99e3b3
              Node         - 0002c9010a99e3b0
              Port 1       - 0002c9010a99e3b1
              Port 2       - 0002c9010a99e3b2
        Firmware revision: 4.8.00c8
        Product           : MTLP25208-CF256T (Lion cub)
        PSID              : MT_00B0000001
```

Alternatively, for a SAS Expander presented as a SCSI Enclosure Services device , we might see output such as this:

```
example# fwflash -l
List of available devices:
Device[0] /devices/pci@0/pci@0/pci@2/scsi@0/ses@3,0:ses
  Class [sgen]
        Target port WWN  : 500605b00002453d
        Vendor           : SUN
        Product          : 16Disk Backplane
        Firmware revision: 5021
```

例 3 Flash Upgrading an IB HCA Device

The following command flash upgrades an IB HCA device.

```
example# fwflash -f ./version.3.2.0000 \
   -d /devices/pci@0,0/pci8086,3597@4/pci15b3,6278@0:devctl
About to update firmware on:
```

例 3 Flash Upgrading an IB HCA Device (続き)

```
  /devices/pci@0,0/pci8086,3597@4/pci15b3,6278@0:devctl
Continue (Y/N): Y

Updating . . . . . . . . . . . .
Done.  New image will be active after the system is rebooted.
```

Note that you are prompted before the upgrading proceeds and that it is mandatory that you reboot your host to activate the new firmware image.

The following command adds the `-y` option to the command.

```
example# fwflash -y -f ./version.3.2.0000 \
   -d /devices/pci@0,0/pci8086,3597@4/pci15b3,6278@0:devctl
About to update firmware on:
  /devices/pci@0,0/pci8086,3597@4/pci15b3,6278@0:devctl

Updating . . . . . . . . . . . .
Done.  New image will be active after the system is rebooted.
```

例 4 Reading Device Firmware to File

The command shown below reads the device firmware to a file. The command uses the `-y` option so that read occurs without prompting.

```
example# fwflash -y -r /firmware.bin \
   -d /devices/pci@1d,700000/pci@1/pci15b3,5a44@0:devctl
About to read firmware on:
         /devices/pci@1d,700000/pci@1/pci15b3,5a44@0:devctl
to filename: /firmware.bin

Reading . . .
Done.
```

例 5 When No Flashable Devices Are Found

The command output shown below informs the user that there are no supported flashable devices found in the system:

```
example# fwflash -l
fwflash: No flashable devices attached with the ses driver in this system
fwflash: No flashable devices attached with the sgen driver in this system
fwflash: No flashable devices attached with the tavor driver in this system
fwflash: No flashable devices in this system
```

Each plugin found in `/usr/lib/fwflash/identify` is loaded in turn, and walks the system device tree, determining whether any currently-attached devices can be flashed. For the list of device types and drivers that are currently supported, please see the `NOTES` section below.

戻り値

The `fwflash` command returns the following values:

`0`
: Success

`1`
: Failure

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/firmware-utilities
Interface Stability	Committed

関連項目

`attributes(5)`, `hermon(7D)`, `ses(7D)`, `tavor(7D)`

The InfiniBand Trade Association website, `http://www.infinibandta.org`

The SCSI Storage Interfaces committee website, `http://www.t10.org`

SCSI Primary Commands-4, SPC4

SCSI Enclosure Services-2, SES2

Serial Attached SCSI-2, SAS2

注意事項

The `fwflash` command supports:

- InfiniBand Host Channel Adapters (IB HCAs) containing either the AMD or the Intel parallel flash parts.
- SCSI Enclosure Services devices such as SAS Expanders, attached with `ses(7D)` drivers.

名前

fwtmp, wtmpfix – manipulate connect accounting records

形式

```
/usr/lib/acct/fwtmp [-ic]
```

```
/usr/lib/acct/wtmpfix [file]...
```

機能説明

fwtmp reads from the standard input and writes to the standard output, converting binary records of the type found in /var/adm/wtmpx to formatted ASCII records. The ASCII version is useful when it is necessary to edit bad records.

wtmpfix examines the standard input or named files in utmpx format, corrects the time/date stamps to make the entries consistent, and writes to the standard output. A hyphen (–) can be used in place of *file* to indicate the standard input. If time/date corrections are not performed, acctcon(1M) will fault when it encounters certain date-change records.

Each time the date is set, a pair of date change records are written to /var/adm/wtmpx. The first record is the old date denoted by the string "old time" placed in the line field and the flag OLD_TIME placed in the type field of the utmpx structure. The second record specifies the new date and is denoted by the string new time placed in the line field and the flag NEW_TIME placed in the type field. wtmpfix uses these records to synchronize all time stamps in the file.

In addition to correcting time/date stamps, wtmpfix will check the validity of the name field to ensure that it consists solely of alphanumeric characters or spaces. If it encounters a name that is considered invalid, it will change the login name to INVALID and write a diagnostic to the standard error. In this way, wtmpfix reduces the chance that acctcon will fail when processing connect accounting records.

オプション

-ic　Denotes that input is in ASCII form, and output is to be written in binary form.

ファイル

/var/adm/wtmpx　history of user access and administration information

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), ed(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), runacct(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

『Oracle Solaris の管理: 一般的なタスク』

名前

getdevpolicy – inspect the system's device policy

形式

`/usr/sbin/getdevpolicy [device...]`

機能説明

Without arguments, `getdevpolicy` outputs the device policy in effect to standard output.

With arguments, each argument is treated as a pathname to a device and the device policy in effect for that specific device is printed preceded by the supplied pathname.

使用法

The device policy adds access restrictions over and above the file permissions.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`non-zero` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below

The invocation is evolving. The output is unstable.

関連項目

`add_drv(1M)`, `rem_drv(1M)`, `update_drv(1M)`, `attributes(5)`, `privileges(5)`, `devfs(7FS)`

名前

getent – get entries from administrative database

形式

`getent` *database* `[`*key*`]...`

機能説明

`getent` gets a list of entries from the administrative database specified by *database*. The information generally comes from one or more of the sources that are specified for the *database* in `/etc/nsswitch.conf`.

database is the name of the database to be examined. This can be `passwd`, `group`, `hosts`, `ipnodes`, `services`, `protocols`, `auth_attr` or `exec_attr`. For each of these databases, `getent` uses the appropriate library routines described in `getpwnam`(3C), `getgrnam`(3C), `gethostbyaddr`(3NSL), `gethostbyname`(3NSL), `getipnodebyaddr`(3SOCKET), `getipnodebyname`(3SOCKET), `getservbyname`(3SOCKET), `getprotobyname`(3SOCKET), `getprofattr`(3C), `getauthattr`(3C), and `getexecattr`(3C), respectively.

Each *key* must be in a format appropriate for searching on the respective database. For example, it can be a *username* or *numeric-uid* for `passwd`; *hostname* or *IP address* for `hosts`; or *service*, *service/protocol*, *port*, or *port/proto* for `services`.

`getent` prints out the database entries that match each of the supplied keys, one per line, in the format of the matching administrative file: `passwd`(4), `group`(4), `project`(4), `networks`(4), `netmasks`(4), `user_attr`(4), `prof_attr`(4), `auth_attr`(4), or `exec_attr`(4). The key for `exec_attr`(4) is a profile name. If no key is given, all entries returned by the corresponding enumeration library routine, for example, `getpwent()` or `gethostent()`, are printed. Enumeration is not supported on `ipnodes`, `ethers`, and `netmasks`.

Key Interpretation for passwd and group Databases

When `getent` is invoked with database set to `passwd`, each key value is processed as follows:

- If the key value consists only of numeric characters, `getent` assumes that the key value is a numeric user ID and searches the user database for a matching user ID.
- If the user ID is not found in the user database or if the key value contains any non-numeric characters, `getent` assumes the key value is a user name and searches the user database for a matching user name.

Similarly, when `getent` is invoked with database set to `group`, each key value is processed as follows:

- If the key value consists only of numeric characters, `getent` assumes that the key value is a numeric group ID and searches the group database for a matching group ID.
- If the group ID is not found in the `group` database or if the key value contains any non-numeric characters, `getent` assumes the key value is a group name and searches the `group` database for a matching group name.

When `getent` is invoked with database set to `user_attr`(4), each key value is processed as follows:

- If the key value consists only of numeric characters, `getent` assumes that the key value is a numeric user ID and searches the user database for a matching user ID.

- If the key value contains any non-numeric characters, `getent` assumes the key value is a user name and searches the user database for a matching user name.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Command syntax was incorrect, an invalid option was used, or an internal error occurred.

`2` At least one of the specified entry names was not found in the database.

`3` There is no support for enumeration on this database.

ファイル

`/etc/nsswitch.conf`	name service switch configuration file
`/etc/passwd`	password file
`/etc/group`	group file
`/etc/inet/hosts`	IPv4 and IPv6 host name database
`/etc/services`	Internet services and aliases
`/etc/project`	project file
`/etc/protocols`	protocol name database
`/etc/ethers`	Ethernet address to hostname database or domain
`/etc/networks`	network name database
`/etc/netmasks`	network mask database
`/etc/user_attr`	Extended user attributes
`/etc/security/prof_attr`	Profile description database
`/etc/security/auth_attr`	Authorization description
`/etc/security/exec_attr`	Execution profiles database

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`ethers(3SOCKET)`, `getgrnam(3C)`, `gethostbyaddr(3NSL)`, `gethostbyname(3NSL)`, `gethostent(3NSL)`, `getipnodebyaddr(3SOCKET)`, `getipnodebyname(3SOCKET)`, `getnetbyname(3SOCKET)`, `getprojbyname(3PROJECT)`, `getprotobyname(3SOCKET)`, `getpwnam(3C)`, `getservbyname(3SOCKET)`, `getauthattr(3C)`, `getexecattr(3C)`,

getprofattr(3C), getuserattr(3C), ethers(4), group(4), hosts(4), netmasks(4), networks(4), nsswitch.conf(4), auth_attr(4), exec_attr(4), passwd(4), prof_attr(4), project(4), protocols(4), services(4), user_attr(4), attributes(5)

名前

gettable – get DoD Internet format host table from a host

形式

`/usr/sbin/gettable` *host*

機能説明

`gettable` is a simple program used to obtain the DoD Internet host table from a "hostname" server. The specified *host* is queried for the table. The table is placed in the file `hosts.txt`.

`gettable` operates by opening a TCP connection to the port indicated in the service specification for "hostname". A request is then made for all names and the resultant information is placed in the output file.

`gettable` is best used in conjunction with the `htable(1M)` program which converts the DoD Internet host table format to that used by the network library lookup routines.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`htable(1M)`, `attributes(5)` Harrenstien, Ken, Mary Stahl, and Elizabeth Feinler, HOSTNAME Server, RFC 953, Network Information Center, SRI International, Menlo Park, California, October 1985.

注意事項

Should allow requests for only part of the database.

名前

getty – set terminal type, modes, speed, and line discipline

形式

```
/usr/lib/saf/ttymon [-h] [-t timeout] line
     [speed [type [linedisc]]]

/usr/lib/saf/ttymon -c file
```

機能説明

`getty` sets terminal type, modes, speed, and line discipline. `getty` is a symbolic link to `/usr/lib/saf/ttymon`. It is included for compatibility with previous releases for the few applications that still call `getty` directly.

`getty` can only be executed by the super-user, (a process with the user ID `root`). Initially `getty` prints the login prompt, waits for the user's login name, and then invokes the `login` command. `getty` attempts to adapt the system to the terminal speed by using the options and arguments specified on the command line.

Without optional arguments, `getty` specifies the following: The *speed* of the interface is set to 300 baud, either parity is allowed, NEWLINE characters are converted to carriage return-line feed, and tab expansion is performed on the standard output. `getty` types the login prompt before reading the user's name a character at a time. If a null character (or framing error) is received, it is assumed to be the result of the user pressing the BREAK key. This will cause `getty` to attempt the next *speed* in the series. The series that `getty` tries is determined by what it finds in `/etc/ttydefs` .

オプション

The following options are supported:

`-h`	If the `-h` flag is not set, a hangup will be forced by setting the speed to zero before setting the speed to the default or a specified speed.
`-t` *timeout*	Specifies that `getty` should exit if the open on the line succeeds and no one types anything in *timeout* seconds.
`-c` *file*	The `-c` option is no longer supported. Instead use `/usr/sbin/sttydefs -l` to list the contents of the `/etc/ttydefs` file and perform a validity check on the file.

オペランド

The following operands are supported:

line	The name of a TTY line in `/dev` to which `getty` is to attach itself. `getty` uses this string as the name of a file in the `/dev` directory to open for reading and writing.
speed	The *speed* argument is a label to a speed and TTY definition in the file `/etc/ttydefs`. This definition tells `getty` at what speed to run initially, what the initial TTY settings are, and what speed to try next, (should the user press the BREAK key to indicate that the speed is inappropriate). The default *speed* is 300 baud.
type and *linedisc*	These options are obsolete and will be ignored.

ファイル

/etc/ttydefs

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

ct(1C), login(1), sttydefs(1M), ttymon(1M), ioctl(2), attributes(5), tty(7D)

名前

gkadmin – Kerberos principals and policies administration GUI

形式

/usr/sbin/gkadmin

機能説明

gkadmin is an interactive graphical user interface (GUI) that enables you to maintain Kerberos principals and policies. gkadmin provides much the same functionality as the kadmin(1M) command.

gkadmin does not support the management of keytabs. You must use kadmin for keytabs management. gkadmin uses Kerberos authentication and an encrypted RPC to operate securely from anywhere on the network.

When gkadmin is invoked, the login window is populated with default values. For the principal name, gkadmin determines your user name from the USER environment variable. It appends /admin to the name (*username*/admin) to create a default user instance in the same manner as kadmin. It also selects appropriate defaults for realm and master KDC (*admin_server*) from the /etc/krb5/krb5.conf file.

You can change these defaults on the login window. When you enter your password, a session is started with kadmind. Operations performed are subject to permissions that are granted or denied to the chosen user instance by the Kerberos ACL file. See kadm5.acl(4).

After the session is started, a tabbed folder is displayed that contains a principal list and a policy list. The functionality is mainly the same as kadmin, with addition, deletion, and modification of principal and policy data available.

gkadmin also includes an interface to specify principal key encryption types when modifying or creating principal records. The default set of encryption types is used if they are not selected through this interface. The default set of encryption types can be found in krb5.conf(4) under the default_tkt_enctypes section.

In addition, gkadmin provides the following features:

- New principal or policy records can be added either from default values or from the settings of an existing principal.
- A comment field is available for principals.
- Default values are saved in $HOME/.gkadmin.
- A logout option permits you to log back in as another user instance without exiting the tool.
- Principal and policy lists and attributes can be printed or saved to a file.
- Online context-sensitive help and general help is available in the Help menu.

ファイル

/etc/krb5/krb5.conf	Kerberos configuration information on a Kerberos client. Used to search for default realm and master KDC (admin_server), including a port number for the master KDC.

`$HOME/.gkadmin`	Default parameters used to initialize new principals created during the session.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

`kpasswd(1)`, `kadmin(1M)`, `kadmind(1M)`, `kadmin.local(1M)`, `kdb5_util(1M)`, `kadm5.acl(4)`, `kdc.conf(4)`, `krb5.conf(4)`, `attributes(5)`, `kerberos(5)`

診断

The `gkadmin` interface is currently incompatible with the MIT `kadmind` daemon interface, so you cannot use this interface to administer an MIT-based Kerberos database. However, clients running the Solaris implementation of Kerberos can still use an MIT-based KDC.

名前

groupadd – add (create) a new group definition on the system

形式

```
/usr/sbin/groupadd [-g gid [-o]] [-S repository]
      [-U user1[,user2..] ] group
```

機能説明

The `groupadd` command creates a new group definition on the system by adding the appropriate entry to the group database in the `files` and `ldap` repositories.

An administrator must be granted the User Management rights profile or have `solaris.group.manage` authorization to be able to add a group. Once the group is successfully added, the administrator is granted the authorization to modify and delete the group. The format of the authorization is `solaris.group.assign`/*groupname*, and can be managed by means of `usermod(1M)`.

オプション

The following options are supported:

`-g` *gid*
: Assigns the group id *gid* for the new group. This group id must be a non-negative decimal integer below `MAXUID` as defined in `/usr/include/sys/param.h`. The group ID defaults to the next available (unique) number above the highest number currently assigned. For example, if groups `100`, `105`, and `200` are assigned as groups, the next default group number is `201`. (Group IDs from `0`–`99` are reserved by SunOS for future applications.)

`-o`
: Allows the *gid* to be duplicated (non-unique). An administrator must have `solaris.group.assign` authorization to use this option.

`-S` *repository*
: The valid repositories are `files` and `ldap`. The repository specifies which name service will be updated. When *repository* is not specified, the `files` repository is used. When the repository is `files`, the user name and other items can be present in other name service repositories and can be assigned to a group in the `files` repository. When the repository is `ldap`, all the assignable attributes must be present in the `ldap` repository.

`-U` *user1*[,*user2*]
: Adds a list of users *user1*, *user2* to the group.

オペランド

The following operands are supported:

group
: A string consisting of characters from the set of lower case alphabetic characters and numeric characters. A warning message is written if the string exceeds `MAXGLEN-1`, which is usually eight characters. The *group* field must contain at least one character; it accepts lower case or numeric characters or a combination of both, and must not contain a colon (`:`) or NEWLINE.

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`2`
: Invalid command syntax. A usage message for the `groupadd` command is displayed.

3 An invalid argument was provided to an option.

4 The *gid* is not unique (when -o option is not used).

9 The *group* is not unique.

10 The group database cannot be updated.

ファイル

- /etc/group
- /usr/include/userdefs.h

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

users(1B), groupdel(1M), groupmod(1M), grpck(1M), logins(1M), pwck(1M), useradd(1M), userdel(1M), usermod(1M), group(4), attributes(5)

注意事項

groupadd adds a group definition to the system. If a network name service is being used to supplement the local /etc/group file with additional entries, groupadd verifies the uniqueness of a specified group name and group ID against the external name service and uses the entries in the files repository.

If the number of characters in a group entry exceeds 2047, group maintenance commands, such as groupdel(1M) and groupmod(1M), fail.

名前

groupdel – delete a group definition from the system

形式

`/usr/sbin/groupdel [-S repository] group`

機能説明

The `groupdel` utility deletes a group definition from the system. It deletes the appropriate entry from the `/etc/group` file.

In addition to `solaris.group.manage` authorization, an administrator must have either `solaris.group.assign` or a matching authorization of the form `solaris.group.assign/`*groupname* to delete a group. The authorization `solaris.group.assign/`*groupname* is automatically assigned to the administrator who created the group.

オプション

`groupdel` supports the following option.

`-S` *repository*
: The valid repositories are `files` and `ldap`. The repository specifies which name service will be updated. When *repository* is not specified, `groupdel` consults `nsswitch.conf(4)`.

オペランド

group An existing group name to be deleted.

終了ステータス

The following exit values are returned:

`0` Success.

`2` Invalid command syntax. A usage message for the `groupdel` command is displayed.

`6` *group* does not exist.

`10` Cannot update the `/etc/group` file.

ファイル

`/etc/group` system file containing group definitions

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`users(1B)`, `groupadd(1M)`, `groupmod(1M)`, `logins(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `nsswitch.conf(4)`, `attributes(5)`

注意事項

The `groupdel` utility deletes a group definition that is in the group database in the repository.

`groupdel` fails if a group entry (a single line in `/etc/group`) exceeds 2047 characters.

名前

groupmod – modify a group definition on the system

形式

```
/usr/sbin/groupmod [-S repository] [-g gid [-o]] [-n name]
     [-U [+|-]user1[,user2]...] group
```

機能説明

The `groupmod` command modifies the definition of the specified group by modifying the appropriate entry in the group database in the repository.

An administrator can modify any group for which it has a matching authorization of the form `solaris.group.assign/`*groupname*. This authorization is automatically assigned to the administrator who created the group. An administrator must have `solaris.group.assign` authorization to modify all other groups.

オプション

The following options are supported:

`-g` *gid*
: Specify the new group ID for the group. This group ID must be a non-negative decimal integer less than `MAXUID`, as defined in `<param.h>`. The group ID defaults to the next available (unique) number above 99. (Group IDs from 0-99 are reserved by SunOS for future applications.)

`-n` *name*
: Specify the new name for the group. The *name* argument is a string of no more than eight bytes consisting of characters from the set of lower case alphabetic characters and numeric characters. A warning message will be written if these restrictions are not met. A future Solaris release may refuse to accept group fields that do not meet these requirements. The *name* argument must contain at least one character and must not include a colon (`:`) or NEWLINE (`\n`).

`-o`
: Allow the *gid* to be duplicated (non-unique). An administrator must have solaris.group.assign authorization to use this option.

`-S` *repository*
: The valid repositories are `files` and `ldap`. The repository specifies which name service will be updated. When *repository* is not specified, `groupmod` consults `nsswitch.conf(4)`. When the repository is `files`, the user name and other items can be present in other name service repositories and can be assigned to a group in the `files` repository. When the repository is `ldap`, all the assignable attributes must be present in the `ldap` repository.

`-U` [+|-]*user1*[,*user2*]
: Updates the list of users for the group as follows:

- A prefix + before the list adds that list to existing users list.
- A prefix - before the list removes each user in the list from the existing users list.
- With no prefix before the list, replaces the existing users list with the new list of users specified.

オペランド

The following operands are supported:

group An existing group name to be modified.

終了ステータス

The `groupmod` utility exits with one of the following values:

`0` Success.

`2` Invalid command syntax. A usage message for the `groupmod` command is displayed.

`3` An invalid argument was provided to an option.

`4` *gid* is not unique (when the `-o` option is not used).

`6` *group* does not exist.

`9` *name* already exists as a group name.

`10` Cannot update the `/etc/group` file.

ファイル

`/etc/group` group file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`users(1B)`, `groupadd(1M)`, `groupdel(1M)`, `logins(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `group(4)`, `nsswitch.conf(4)`, `attributes(5)`

注意事項

The `groupmod` utility only modifies group definitions in the group database in the repository. If a network name service such as NIS is being used to supplement the local `/etc/group` file with additional entries, `groupmod` cannot change information supplied by the network name service. `groupmod` verifies the uniqueness of group name and group ID against the external name service and uses the entries in the `files` repository.

`groupmod` fails if a group entry (a single line in `/etc/group`) exceeds 2047 characters.

名前

growfs – non-destructively expand a UFS file system

形式

```
/usr/sbin/growfs [-M mount-point] [newfs-options]
      [raw-device]
```

機能説明

`growfs` non-destructively expands a mounted or unmounted UNIX file system (UFS) to the size of the file system's slice(s).

Typically, disk space is expanded by first adding a slice to a metadevice, then running the `growfs` command. When adding space to a mirror, you expand each submirror before expanding the file system.

`growfs` will "write-lock" (see `lockfs(1M)`) a mounted file system when expanding. The length of time the file system is write-locked can be shortened by expanding the file system in stages. For instance, to expand a 1 Gbyte file system to 2 Gbytes, the file system can be grown in 16 Mbyte stages using the `-s` option to specify the total size of the new file system at each stage. The argument for `-s` is the number of sectors, and must be a multiple of the cylinder size. Note: The file system cannot be grown if a cylinder size of less than 2 is specified. Refer to the `newfs(1M)` man page for information on the options available when growing a file system.

`growfs` displays the same information as `mkfs` during the expansion of the file system.

If `growfs` is aborted, recover any lost free space by unmounting the file system and running the `fsck` command, or run the `growfs` command again.

Note: If `growfs` is aborted and the file system is used before `fsck` is run on it, UFS metadata might be left in an incomplete state, with the result that the file system would be corrupted. In such a circumstance, you would have to restore the file system from backups.

オプション

Root privileges are required for all of the following options.

`-M` *mount-point*	The file system to be expanded is mounted on *mount-point*. File system locking (`lockfs`) will be used.
newfs-options	The options are documented in the `newfs` man page.
raw-device	Specifies the name of a raw metadevice or raw special device, residing in `/dev/md/rdsk`, or `/dev/rdsk`, respectively, including the disk slice, where you want the file system to be grown.

使用例

例 1　Expanding nonmetadevice slice for /export file system

The following example expands a nonmetadevice slice for the `/export` file system. In this example, the existing slice, `/dev/dsk/c1t0d0s3`, is converted to a metadevice so additional slices can be concatenated.

```
# metainit -f d8 2 1 c1t0d0s3 1 c2t0d0s3
# umount /export
```

例 2 Associate /export with new metadevice

Edit the /etc/vfstab file to change the entry for /export to the newly defined metadevice, d8.

```
# mount /export
# growfs -M /export /dev/md/rdsk/d8
```

The first example starts by running the metainit command with the -f option to force the creation of a new concatenated metadevice d8, which consists of the existing slice /dev/dsk/c1t0d0s3 and a new slice /dev/dsk/c2t0d0s3. Next, the file system on /export must be unmounted. The /etc/vfstab file is edited to change the entry for /export to the newly defined metadevice name, rather than the slice name. After the file system is remounted, the growfs command is run to expand the file system. The file system will span the entire metadevice when growfs completes. The -M option enables the growfs command to expand a mounted file system. During the expansion, write access for /export is suspended until growfs unlocks the file system. Read access is not affected, though access times are not kept when the lock is in effect.

例 3 Dynamic Expansion of /export file system

The following example picks up from the previous one. Here, the /export file system mounted on metadevice d8 is dynamically expanded.

```
# metattach d8 c0t1d0s2
# growfs -M /export /dev/md/rdsk/d8
```

This example begins by using the metattach command to dynamically concatenate a new slice, /dev/dsk/c0t1d0s2, to the end of an existing metadevice, d8. Next, the growfs command specifies that the mount-point is /export and that it is to be expanded onto the raw metadevice /dev/md/rdsk/d8. The file system will span the entire metadevice when growfs completes. During the expansion, write access for /export is suspended until growfs unlocks the file system. Read access is not affected, though access times are not kept when the lock is in effect.

例 4 Expanding mounted file system to existing mirror

The following example expands a mounted file system /files, to an existing mirror, d80, which contains two submirrors, d9 and d10.

```
# metattach d9 c0t2d0s5
# metattach d10 c0t3d0s5
# growfs -M /files /dev/md/rdsk/d80
```

In this example, the metattach command dynamically concatenates the new slices to each submirror. The metattach command must be run for each submirror. The mirror will automatically grow when the last submirror is dynamically concatenated. The mirror will grow to the size of the smallest submirror. The growfs command then expands the file system. The growfs command specifies that the mount-point is /files and that it is to be expanded

例 4 Expanding mounted file system to existing mirror (続き)

onto the raw metadevice /dev/md/rdsk/d80. The file system will span the entire mirror when the growfs command completes. During the expansion, write access for the file system is suspended until growfs unlocks the file system. Read access is not affected, though access times are not kept when the lock is in effect.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm

関連項目

fsck(1M), lockfs(1M), mkfs(1M), metattach(1M), newfs(1M), attributes(5)

Solaris Volume Manager Administration Guide

Limitations

Only UFS file systems (either mounted or unmounted) can be expanded using the growfs command. Once a file system is expanded, it cannot be decreased in size. The following conditions prevent you from expanding file systems: When acct is activated and the accounting file is on the target device. When C2 security is activated and the logging file is on the target file system. When there is a local swap file in the target file system. When the file system is root (/), /usr, or swap.

名前

gsscred – add, remove, and list gsscred table entries

形式

```
gsscred [-n user [-o oid] [-u uid]] [-c comment] -m mech -a
gsscred [-n user [-o oid]] [-u uid] [-m mech] -r
gsscred [-n user [-o oid]] [-u uid] [-m mech] -l
```

機能説明

The `gsscred` utility is used to create and maintain a mapping between a security principal name and a local UNIX *uid*. The format of the user name is assumed to be `GSS_C_NT_USER_NAME`. You can use the `-o` option to specify the object identifier of the *name* type. The `OID` must be specified in dot-separated notation, for example: `1.2.3.45464.3.1`

The `gsscred` table is used on server machines to lookup the *uid* of incoming clients connected using `RPCSEC_GSS`.

When adding users, if no *user* name is specified, an entry is created in the table for each user from the `passwd` table. If no *comment* is specified, the `gsscred` utility inserts a comment that specifies the user name as an ASCII string and the GSS-APIsecurity mechanism that applies to it. The security mechanism will be in string representation as defined in the `/etc/gss/mech` file.

The parameters are interpreted the same way by the `gsscred` utility to delete users as they are to create users. At least one of the following options must be specified: `-n`, `-u`, or `-m`. If no security mechanism is specified, then all entries will be deleted for the user identified by either the *uid* or *user* name. If only the security mechanism is specified, then all *user* entries for that security mechanism will be deleted.

Again, the parameters are interpreted the same way by the `gsscred` utility to search for users as they are to create users. If no options are specified, then the entire table is returned. If the *user* name or *uid* is specified, then all entries for that *user* are returned. If a security mechanism is specified, then all *user* entries for that security mechanism are returned.

オプション

`-a`	Add a table entry.
`-c` *comment*	Insert comment about this table entry.
`-l`	Search table for entry.
`-m` *mech*	Specify the mechanism for which this name is to be translated.
`-n` *user*	Specify the optional principal name.
`-o` *oid*	Specify the OID indicating the name type of the user.
`-r`	Remove the entry from the table.
`-u` *uid*	Specify the *uid* for the *user* if the *user* is not local.

使用例

例 1 Creating a gsscred Table for the Kerberos v5 Security Mechanism

The following shows how to create a `gsscred` table for the kerberos v5 security mechanism. `gsscred` obtains *user* names and *uid*'s from the `passwd` table to populate the table.

```
example% gsscred -m kerberos_v5 -a
```

例 2 Adding an Entry for root/host1 for the Kerberos v5 Security Mechanism

The following shows how to add an entry for `root/host1` with a specified *uid* of `0` for the kerberos v5 security mechanism.

```
example% gsscred -m kerberos_v5 -n root/host1 -u 0 -a
```

例 3 Listing All User Mappings for the Kerberos v5 Security Mechanism

The following lists all user mappings for the kerberos v5 security mechanism.

```
example% gsscred -m kerberos_v5 -l
```

例 4 Listing All Mappings for All Security Mechanism for a Specified User

The following lists all mappings for all security mechanisms for the user `bsimpson`.

```
example% gsscred -n bsimpson -l
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/security/gss
Interface Stability	Committed

関連項目

`gssd(1M)`, `gsscred.conf(4)`, `attributes(5)`

注意事項

Some GSS mechanisms, such as `kerberos_v5`, provide their own authenticated-name-to-local-name (uid) mapping and thus do not usually have to be mapped using `gsscred`. See `gsscred.conf(4)` for more information.

名前

gssd – generates and validates GSS-API tokens for kernel RPC

形式

/usr/lib/gss/gssd

機能説明

gssd is the user mode daemon that operates between the kernel rpc and the Generic Security Service Application Program Interface (GSS-API) to generate and validate GSS-API security tokens. In addition, gssd maps the GSS-API principal names to the local user and group ids. By default, all groups that the requested user belongs to will be included in the grouplist credential. gssd is invoked by the Internet daemon inetd(1M) the first time that the kernel RPC requests GSS-API services.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/kernel/security/gss
Interface Stability	Committed

関連項目

kill(1), pkill(1), svcs(1), inetadm(1M), inetd(1M), gsscred(1M), svcadm(1M), gsscred.conf(4), resolv.conf(4), attributes(5), smf(5)

RFC 2078

注意事項

The following signal has the specified effect when sent to the server process using the kill(1) command:

SIGHUP gssd rereads the gsscred.conf(4) options.

When one of the mechanisms being used is Kerberos, then the gssd process must be restarted after adding or changing the resolv.conf(4) file.

The gssd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rpc/gss:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

hald – ハードウェア抽象化レイヤーをサポートするデーモン

形式

```
/usr/lib/hal/hald [--daemon={yes | no}]] [--help] [--use-syslog]
 [--verbose={yes | no}] [--version]
```

機能説明

hald デーモンは、ハードウェア抽象化レイヤー (HAL) 仕様に準拠したデバイスのハードウェア変更の認識をサポートします。

hald の有効化や無効化を実行するには、サービス管理機能 (SMF) を使用します (smf(5) を参照)。hald は、障害管理リソース識別子 (FMRI) svc:/system/hal を使用して管理されます。

オプション

次のオプションがサポートされています。

--daemon=yes|no	デーモンとして実行します。
--help	使用法の情報を表示して終了します。
--use-syslog	stderr ではなく syslog にデバッグメッセージを表示します。このオプションは、HAL をデーモンとして実行する場合にデバッグメッセージを記録するために使用します。
--verbose=yes|no	デバッグ情報を表示します。
--version	バージョン情報を表示して終了します。

ファイル

/usr/lib/hal	HAL 関連のファイル
/etc/hal/fdi	デバイス情報ファイル

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/hal
インタフェースの安定性	流動的

関連項目

svccfg(1M), attributes(5), hal(5)

名前

hal-device – HAL デバイスの管理

形式

`hal-device [-h] [--a udi | --r udi]`

機能説明

ハードウェア抽象化レイヤー (HAL) は、システムに接続されている各種ハードウェアのビューを提供します。`hal-device` コマンドを使用すると、HAL 標準に準拠したデバイスを管理できます。具体的には、`hal-device` を使用すると、HAL グローバルデバイスリストからのデバイスの追加、およびそのリストからのデバイスの削除を実行できます。デバイスプロパティーは、標準入力から `lshal(1M)` 構文で読み取られます。

オプション

次のオプションがサポートされています。

`-a`, `--add` *udi*	汎用デバイス識別子 *udi* で指定されたデバイスを、HAL のグローバルデバイスリストに追加します。
`-h`	使用法に関する情報を表示します。
`-r`, `--remove` *udi*	汎用デバイス識別子 *udi* で指定されたデバイスを、HAL のグローバルデバイスリストから削除します。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/hal
インタフェースの安定性	流動的

関連項目

`hald(1M)`, `attributes(5)`, `hal(5)`

名前

hal-fdi-validate – HAL デバイス情報ファイルの検証

形式

`hal-fdi-validate [-f dtd] file [file]...`

機能説明

`hal-fdi-validate` コマンドは、1 つ以上のデバイス情報ファイルを検証します。`fdi(4)` を参照してください。`-f` オプションを使用して別のファイルを指定しないかぎり、標準の DTD ファイルが使用されます。

オプション

次のオプションがサポートされています。

`-f` *dtd*　DTD ファイルへのパスを指定します。

オペランド

`hal-fdi-validate` コマンドで使用できるオペランドは、次のとおりです。

file [*file*...]　検証対象となる 1 つ以上の DTD ファイル。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/hal
インタフェースの安定性	流動的

関連項目

`hald(1M)`, `fdi(4)`, `attributes(5)`, `hal(5)`

名前

hal-find, hal-find-by-capability, hal-find-by-property – HAL グローバルデバイスリストの検索

形式

```
hal-find-by-capability --capability capability [--help]
 [--verbose] [--version]

hal-find-by-property  --key key --string value [--help]
 [--verbose] [--version]
```

機能説明

`hal-find` コマンド `hal-find-by-capability` と `hal-find-by-property` は、指定された条件に基づいてハードウェア抽象化レイヤー (HAL) デバイスリストの検索を行い、結果を標準出力に表示します。`hal-find-by-capability` は、`volume` や `block` などの機能に基づいて検索を行います。`hal-find-by-property` は、`block.is_volume` や `volume.disc.has_audio` などのプロパティーに基づいて検索を行います。

オプション

次のオプションがサポートされています。

`--capability` *capability*	検索対象となる HAL デバイス機能。
`--help`	オプションの一覧を表示します。
`--key` *key*	検索の基準となるプロパティーのキー。
`--string` *value*	検索の基準となるプロパティーに関連付けられた文字列 *value*。
`--verbose`	冗長モード。
`--version`	バージョンを表示して終了します。

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/hal
インタフェースの安定性	流動的

関連項目

`hald(1M)`, `attributes(5)`, `hal(5)`

名前

hal-get-property, hal-set-property – HAL デバイスプロパティーの取得と設定

形式

```
hal-get-property --udi udi --key key [--help] [--verbose]
 [--version]
```

```
hal-set-property  --udi udi --key key {--int value |  --uint64 value
 |  --string value |  --bool value |  --strlist-pre value
 |  --strlist-post value |  --strlist-rem value |  --double value
 |  --remove value} [--help] [--version]
```

機能説明

ハードウェア抽象化レイヤー (HAL) は、システムに接続されている各種ハードウェアのビューを提供します。このビューは、ハードウェア構成に変化があると、ホットプラグなどのメカニズムを通じて動的に更新されます。HAL は、ハードウェアの断片をデバイスオブジェクトとして表します。デバイスオブジェクトは、一意の識別子によって識別され、デバイスプロパティーと呼ばれる一連のキーと値のペアを保持します。プロパティーには、実際のハードウェアから取得されるもの、デバイス情報ファイル (`.fdi` ファイル) からマージされるもの、実際のデバイス構成に関連しているものがあります。

`hal-get-property` および `hal-set-property` コマンドを使用すると、HAL 仕様に準拠したハードウェアのプロパティーを取得および設定できます。

オプション

次のオプションがサポートされています。

`--udi` *udi*
一意のデバイス ID。

`--key` *key*
設定するプロパティーのキー。

`--int`
値を整数に設定します。10 進数値、または先頭に `0x` または `x` の付いた 16 進値のいずれかを指定できます。

`--uint64`
値を整数に設定します。10 進数値、または先頭に `0x` または `x` の付いた 16 進値のいずれかを指定できます。

`--string` *value*
値を文字列に設定します。

`--double` *value*
値を浮動小数点数に設定します。

`--boolean` *value*
値をブール値、つまり true または false に設定します

`--strlist-pre` *value*
リストの先頭に文字列を追加します。

`--strlist-post` *value*
リストの末尾に文字列を追加します。

`--strlist-rem` *value*
リストから文字列を削除します。

`--remove` *value*
このプロパティーを削除するべきであることを示します。

`--version`
バージョンを表示して終了します。

`--help`
オプションの一覧を表示して終了します

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/hal
インタフェースの安定性	流動的

関連項目

`hald(1M)`, `attributes(5)`, `hal(5)`

名前

halt, poweroff – プロセッサの停止

形式

```
/usr/sbin/halt [-dlnqy]

/usr/sbin/poweroff [-dlnqy]
```

機能説明

halt および poweroff ユーティリティは、保留されている情報をディスクに書き出してから、プロセッサを停止させます。poweroff ユーティリティは、可能であれば、マシンの電源を自動的に切断します。

halt および poweroff ユーティリティーは通常、システムシャットダウンをシステムログデーモン syslogd(1M) に送信し、ログインアカウントファイル /var/adm/wtmpx にシャットダウン情報を記録します。これらの処理を抑制するには、-n または -q オプションを指定します。

オプション

次のオプションがサポートされています。

- -d　リブートする前に、システムクラッシュダンプを行います。システムクラッシュダンプの設定については、dumpadm(1M) を参照してください。
- -l　halt を実行したユーザーに関するメッセージを、システムログデーモン syslogd(1M) に送信しないようにします。
- -n　停止前の sync(1M) を行いません。
- -q　即時停止。通常のシャットダウンを実行しません。
- -y　ダイヤルアップ端末からでも、システムを停止します。

ファイル

/var/adm/wtmpx　ユーザーアクセスと管理情報の履歴

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

dumpadm(1M), init(1M), reboot(1M), shutdown(1M), sync(1M), syslogd(1M), inittab(4), attributes(5), smf(5)

注意事項

halt および poweroff ユーティリティーは、smf(5) サービスを正常にシャットダウンしません。また、/etc/rc*num*.d のスクリプトや inittab(4) のシャットダウン処理も実行しません。システムサービスを完全にシャットダウンするためには、shutdown(1M) または init(1M) を使用して Solaris システムをリブートしてください。

名前

hextoalabel – 内部テキストラベルの人間が読める形式への変換

形式

`/usr/sbin/hextoalabel [`*internal-text-sensitivity-label*`]`

`/usr/sbin/hextoalabel -c [`*internal-text-clearance*`]`

機能説明

`hextoalabel` は、内部テキストラベルを人間が読める形式に変換し、結果を標準出力ファイルに書き込みます。多くの場合、内部形式は 16 進数で記述されます。オプションが指定されない場合、ラベルは機密ラベルと見なされます。

内部テキストラベルを指定しない場合、標準入力ファイルからラベルが読み取られます。このコマンドは、内部データベース内に格納されているラベルを緊急に修復する場合に使用できます。

オプション

`-c` 内部テキストラベルを認可上限として識別します。

終了ステータス

次の終了値が返されます。

`0` 成功の場合。

`1` エラーが発生しました。診断は標準エラーファイルに書き込まれます。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

コマンド出力は、同一の `label_encodings` ファイルを使用する場合は「確実」です。コマンド呼び出しは、DIA MAC ポリシーを実装するシステムでは「確実」です。

ファイル

`/etc/security/tsol/label_encodings`
ラベルエンコーディングファイルには、このシステムの定義済みラベルに対する格付け名、語句、制約、および値が含まれます。

関連項目

`atohexlabel(1M)`, `label_to_str(3TSOL)`, `str_to_label(3TSOL)`, `label_encodings(4)`, `attributes(5)`

『Trusted Extensions 構成と管理』

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合のみです。

このファイルは、米国国防情報局 (DIA) の必須アクセス制御 (MAC) ポリシーの一部です。このファイルは、Solaris Trusted Extensions ソフトウェアの将来のリリース向けに作成されるその他の MAC ポリシーには適用されない可能性があります。

名前

host – DNS lookup utility

形式

```
host [-aCdilmrsTvw] [-c class] [-N ndots] [-R number]
     [-t type] [-W wait]  [-4 | -6] name [server]
```

機能説明

The `host` utility performs simple DNS lookups. It is normally used to convert names to IP addresses and IP addresses to names. When no arguments or options are given, `host` prints a short summary of its command line arguments and options.

The *name* argument is the domain name that is to be looked up. It can also be a dotted-decimal IPv4 address or a colon-delimited IPv6 address, in which case `host` by default performs a reverse lookup for that address. The optional *server* argument is either the name or IP address of the name server that `host` should query instead of the server or servers listed in `/etc/resolv.conf`.

オプション

The following options are supported:

`-4`
: Use only IPv4 transport. By default, both IPv4 and IPv6 transports can be used. Options `-4` and `-6` are mutually exclusive.

`-6`
: Use only IPv6 transport. By default, both IPv4 and IPv6 transports can be used. Options `-4` and `-6` are mutually exclusive.

`-a`
: Equivalent to setting the `-v` option and asking `host` to make a query of type `ANY`.

`-c` *class*
: Make a DNS query of class *class*. This can be used to lookup Hesiod or Chaosnet class resource records. The default class is IN (Internet).

`-C`
: Attempt to display the SOA records for zone *name* from all the listed authoritative name servers for that zone. The list of name servers is defined by the NS records that are found for the zone.

`-d`
: Generate verbose output. This option is equivalent to `-v`. These two options are provided for backward compatibility. In previous versions, the `-d` option switched on debugging traces and `-v` enabled verbose output.

`-i`
: Specifies that reverse lookups of IPv6 addresses should use the IP6.INT domain as defined in RFC 1886. The default is to use RFC 3152 domain IP6.ARPA.

`-l`
List mode. This option makes `host` perform a zone transfer for zone *name*, displaying the NS, PTR and address records (A/AAAA). If combined with `-a`, all records will be displayed. The argument is provided for compatibility with previous implementations. Options `-la` is equivalent to making a query of type `AXFR`.

`-m`
Sets the memory usage debugging flags: record, usage, and trace.

`-N` *ndots*
Set the number of dots that have to be in *name* for it to be considered absolute. The default value is that defined using the ndots statement in `/etc/resolv.conf`, or 1 if no ndots statement is present. Names with fewer dots are interpreted as relative names and will be searched for in the domains listed in the `search` or `domain` directive in `/etc/resolv.conf`.

`-r`
Make a non-recursive query. Setting this option clears the `RD` (recursion desired) bit in the query made by `host`. The name server receiving the query does not attempt to resolve *name*. The `-r` option enables `host` to mimic the behaviour of a name server by making non-recursive queries and expecting to receive answers to those queries that are usually referrals to other name servers.

`-R` *number*
Change the number of UDP retries for a lookup. The *number* argument indicates how many times `host` will repeat a query that does not get answered. The default number of retries is 1. If *number* is negative or zero, the number of retries will default to 1.

`-s`
Specifies that the host not send the query to the next name server if any server responds with a `SERVFAIL` response, which is the reverse of normal stub resolver behavior.

`-t` *type*
Select the query type. The *type* argument can be any recognised query type: `CNAME`, `NS`, `SOA`, `SIG`, `KEY`, and `AXFR`, among others. When no query type is specified, `host` automatically selects an appropriate query type. By default it looks for A, AAAA, and MX records, but if the `-C` option is specified, queries are made for SOA records. If *name* is a dotted-decimal IPv4 address or colon-delimited IPv6 address, `host` queries for PTR records.

If a query type of IXFR is chosen the starting serial number can be specified by appending an equal followed by the starting serial number (for example: `-t IXFR=12345678`).

`-T`
Use a TCP connection when querying the name server. TCP is automatically selected for queries that require it, such as zone transfer (`AXFR`) requests. By default `host` uses UDP when making queries.

`-v`
Generate verbose output. This option is equivalent to `-d`.

`-w`
Wait forever for a reply. The time to wait for a response will be set to the number of seconds given by the hardware's maximum value for an integer quantity.

`-W` *wait*
Wait for *wait* seconds for a reply. If *wait* is less than one, the wait interval is set to one second.

ファイル

`/etc/resolv.conf`
Resolver configuration file

属性

See for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/dns/bind
Interface Stability	Volatile

関連項目

`dig(1M)`, `named(1M)`, `attributes(5)`

RFC 1035, RFC 1886, RFC 3152

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at `https://www.isc.org/software/bind/documentation`.

名前

hostconfig – configure a system's host parameters

形式

```
/usr/sbin/hostconfig -p protocol [-d] [ -h] [-n] [-v]
     [-i interface] [-f hostname]
```

機能説明

The `hostconfig` program uses a network protocol to acquire a machine's *host parameters* and set these parameters on the system.

The program selects which protocol to use based on the argument to the required `-p` flag. Different protocols may set different host parameters. Currently, only one protocol (`bootparams`) is defined.

オプション

The following options are supported:

`-d`	Enable `debug` output.
`-f` *hostname*	Run the protocol as if this machine were named `hostname`.
`-h`	Echo the received `hostname` to `stdout`, rather than setting `hostname` using the system name directly.
`-i` *interface*	Use only the named network interface to run the protocol.
`-n`	Run the network protocol, but do not set the acquired parameters into the system.
`-p` *protocol*	Run `hostconfig` using *protocol*. Currently, only one protocol (`bootparams`) is available. This option is required. Specifying the `-p bootparams` option uses the `whoami` call of the RPC `bootparams` protocol. This sets the system's `hostname`, `domainname`, and default IP router parameters.
`-v`	Enable verbose output.

使用例

例 1 Configuring Host Parameters with Verbose Output

The following command configures a machine's host parameters using the `whoami` call of the RPC `bootparams` protocol with a verbose output.

```
example# hostconfig -p bootparams -v
```

例 2 Displaying Host Parameters

The following command displays the parameters that would be set using the `whoami` call of the RPC `bootparams` protocol.

```
example# hostconfig -p bootparams -n -v
```

例 3 Configuring Host Parameters Less the System Name

The following command configures a machine's host parameters, less the system name, using the `whoami` call of the RPC `bootparams` protocol.

```
example# hostconfig='hostconfig -p bootparams -h'
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`hostname(1)`, `domainname(1M)`, `route(1M)`, `attributes(5)`

名前

hotplug – configure hotplug connectors and ports

形式

```
hotplug list [-c] [-d] [-l] [-v] [path [connection]]
hotplug online path port
hotplug offline [-f] [-q] path port
hotplug enable [path] connector
hotplug disable [-f] [-q] [path] connector
hotplug poweron [path] connector
hotplug poweroff [-f] [-q] [path] connector
hotplug set -o options [path] connector
hotplug get -ooptions [path] connector
hotplug install path port
hotplug uninstall [-f] [-q] path port
hotplug -?
```

機能説明

The `hotplug` command is used to manage hotplug connections. A connection can be a connector or port. A hotplug connector is a representation of a physical point in the system where components can be inserted or removed. A hotplug port is a representation of a logical point in the system device tree where the connection of a device to the system is managed.

The `hotplug` command only supports hotplug operations on hotplug connectors for PCI Express buses and PCI buses that implement the Standard PCI Hotplug feature. Hotplug ports on PCI Expresss and PCI buses in systems with PCI Express fabrics are also supported. Additional buses may be supported in the future.

The `hotplug` command operates on the following kinds of objects:

`path`

Hotplug connectors and ports are integrated into the Solaris device tree. The names of connectors and ports are unique relative only to their bus controller. A device path is required to uniquely reference a connector and the parent path of the device is required to uniquely reference a port.

The `hotplug` command can accept the case that a user specifies only a physical hotplug connector for the state-change operation subcommands. If no such connector exists, the command will fail. If multiple connectors exist with the same name in the system, the state change operations interact with the user to determine which connector needs to be operated upon. See the “Examples” section, below.

`connector`

If a hardware component supports being physically inserted or removed, then a hotplug connector represents the location where this action may occur. When a connector exists, it has a hierarchy of ports and device nodes that depend upon it.

`port`
: All device nodes can be virtually hotplugged, even if their hardware does not support physical hotplugging. A hotplug port exists between a device node and its parent node in the system device tree. It represents the location where the device node and its dependents can be managed.

`connection`
: A hotplug connection is a generic term to refer to either a hotplug connector or a hotplug port.

Hotplug connectors and ports are managed according to a state model. The `hotplug` command can list information about the hotplug connections in a system, or it can initiate change of state operations on specific hotplug connections.

Hotplug connectors can be in the following states:

`empty`
: A component is not physically inserted in the connector.

`present`
: A component is physically inserted in the connector, but the component is powered off. The component is not in use.

`powered`
: A component is physically inserted in the connector, and the component is powered on. The component is disabled and is not in use.

`enabled`
: A component is physically inserted in the connector. The component is powered on and has been probed and tested. The component is enabled and devices that represent its functions can be used.

Hotplug ports can be in the following states:

`port-empty`
: No device exists for the hotplug port.

`port-present`
: A device exists for the hotplug port, but the device has not been probed and it has no attached device driver. The device is not in use.

`offline`
: A device exists for the hotplug port, and the device has been probed. A device driver is not attached, and the device is not in use.

`online`
: A device exists for the hotplug port, and its device driver is fully attached. The device is in use.

`maintenance`
: A device exists for the hotplug port, and its device driver is fully attached. The device is in use, but not fully operational. A maintenance or fault management operation is affecting the device. The reason that caused the device to enter maintenance state may vary. It is described by a sub-state under maintenance state. Currently there is only one sub-state defined:

  `maintenance-suspended`
  : The device is live suspended.

The `hotplug` command can also access bus private properties for each hotplug connector. The current values of bus private properties can be displayed. New values for each bus private property can be set directly.

サブコマンド

The following subcommands are supported:

`list`
: Show information for hotplug connectors, ports, and their associated devices. Hotplug connectors and hotplug ports are integrated into the Solaris device tree hierarchy. The `list` subcommand therefore displays the hierarchy of device nodes with additional information included to show the locations of hotplug connectors and hotplug ports. The names of hotplug connectors are enclosed in square brackets, and the names of hotplug ports are enclosed in angled brackets. The current state of each hotplug connection is displayed next to its name.

`online`
: Change the state of a hotplug port to the `online` state.

`offline`
: Change the state of a hotplug port to the `offline` state.

`enable`
: Change the state of a hotplug connector to the `enabled` state. All of the hotplug connector's dependent ports will be automatically probed and initialized into the `online` state.

`disable`
: Change the state of a hotplug connector from the `enabled` state to the `powered` state. All dependent ports that are in the `online` state will first be transitioned to the `port-present` state.

`poweron`
: Change the state of a hotplug connector from the `present` state to the `powered` state.

`poweroff`
: Change the state of a hotplug connector from the `powered` or `enabled` state to the `present` state. All dependent ports that are in the `online` state will first be transitioned to the `port-present` state, and will then be removed.

`set`
: Set bus-specific properties for a hotplug connector. The specified option string is a bus specific string of name and value pairs, as could be parsed by `getsubopt(3C)`. The names and values will be passed directly to the bus controller that manages the specified hotplug connector to perform a bus-specific function.

`get`
: Display the current values of bus specific properties for a hotplug connector. The specified option string is a bus specific string of named properties, as could be parsed by `getsubopt(3C)`. The names will be passed directly to the bus controller to specify which properties should be returned. The current values of each named property will then be displayed.

The `install` and `uninstall` subcommands install and uninstall services the drivers of the ports can support.

`install`
: The `install` subcommand install services the drivers of the hotplug port's device can support.

  For example, this subcommand can be applied to the port of physical function of PCIe IO virtualization devices. It upgrades the port to `ONLINE` state (if it is not yet in that state) and then installs the virtual functions that the physical function (driver) supports. New hotplug ports will be created to represent each virtual function as a dependent of the specified physical function. The newly created ports will be initiated to `OFFLINE` state.

`uninstall`
: The `uninstall` subcommand uninstall services the drivers of the hotplug port's device can support.

  This subcommand can be applied to the ports of physical functions of PCIe IO virtualization devices. If the specified hotplug port has any dependent ports of virtual functions, the dependent ports and corresponding virtual function nodes will be removed.

オプション

The following options are supported:

`-l`, `--list-path`
: Show full paths to connections and device nodes. By default, the `list` subcommand shows hotplug connectors, ports, and devices in the format of a tree. This option enables the display of full paths to each connection and device node.

`-c`, `--connectors`
: Display a table that summarizes the current status of all physical hotplug connectors. Device topologies and hierachical information are not included. In general, the names of physical hotplug connectors should be unique. If multiple connectors of the same name exist in the system, `hotplug` displays a message specifying the same-named hotplug connectors and prompts for information to distinguish among those connectors. See “Examples,” below.

`-d`, `--drivers`
Show the binding driver names and the instance number of the device nodes. By default, the `list` subcommand shows only hotplug connectors, ports, and devices. This option enables the display of the binding driver names and the instance numbers of the device nodes.

`-v`, `--verbose`
Show verbose usage details. By default, the `list` subcommand shows only hotplug connectors, ports, and devices. This option enables the display of more detailed information about how the devices are currently consumed. Examples include mounted filesystems or plumbed network interfaces associated with individual devices.

Note that the `-v` option does not display information for disks under ZFS control.

`-f`, `--force`
Force the operation. Some change state operations that impact resources currently in use will fail with a warning. A forced operation will attempt to ignore these warnings and proceed.

This option should be used with extreme caution.

`-n`, `--non-interactive`
Disable the interactive feature. If the flag is specified and an ambiguous input is encountered, the command will exit immediately, returning a unique exit status to indicate the problem.

`-q`, `--query`
Query the operation. Instead of actually performing a change state operation, perform a test to predict if the operation would succeed or fail. If it would fail, show the error messages that would be expected if the operation had really been attempted.

It is not possible to predict every failure. An operation that succeeds during a query could still fail for another reason when actually attempted.

This option will not actually change the state of the system.

`-o` *options*, `--=`*options*
Specify bus-specific properties for a `set` or `get` command. The options string conforms to the `getsubopt(3C)` syntax convention.

For the `get` subcommand, there are two special options that can be used. The special options value of `help` will display all supported properties and their possible values. The special options value of `all` will display the current value of all supported properties.

For the `set` subcommand, there is one special option that can be used. The special options value of `help` will display all supported properties which can be set and their possible values.

See “Notes” section for the properties supported by bus controllers.

`-?, --help`
Display a brief help message on proper use of the command.

使用例

例 1 Showing All Hotplug Connections

The following command shows all hotplug connections:

```
# hotplug list -v
pci@0,0
pci108e,534a@2,1 <pci.2,1> ONLINE
        [pci30] EMPTY
pci10de,5d@e <pci.e,0> ONLINE
        display@b <pci.b,0> ONLINE
        [NEM0] ENABLED
        pci108e,534a@a,0 <pci.a,0> ONLINE
                { Network interface nge0 }
                { nge0: hosts IP addresses: 10.0.0.1 }
        pci108e,534a@a,1 <pci.a,1> (MAINTENANCE)
                           [NEM1] (EMPTY)
pci108e,534a@c <pci.c,0> OFFLINE
pci108e,534a@d <pci.d,0> ONLINE
                          pci1028,40d@0 <pci.0,0> (MAINTENANCE-SUSPENDED,
"activities=dma+pio+intr,reason=resource-rebalance")
                                 { Network interface bge0 }
                                 { bge0: hosts IP addresses: 10.0.1.1 }
```

To show the full paths of hotplug connections and devices, enter the following command:

```
# hotplug list -l
/pci@0,0
/pci@0,0/pci108e,4341  <pci.0,0>  OFFLINE
/pci@0,0/pci8086,3408  <pci.1,0>  OFFLINE
/pci@0,0/pci8086,340a@3  <pci.3,0>  ONLINE
/pci@0,0/pci8086,340a@3 [Slot2]  EMPTY
/pci@0,0/pci8086,340c@5  <pci.5,0>  ONLINE
/pci@0,0/pci8086,340c@5 [Slot1]  ENABLED
/pci@0,0/pci8086,340c@5/pci111d,8018@0  <pci.0,0>  ONLINE
/pci@0,0/pci8086,340e@7  <pci.7,0>  ONLINE
/pci@0,0/pci8086,340e@7 [pcie4]  ENABLED
/pci@0,0/pci8086,3410@9  <pci.9,0>  ONLINE
/pci@0,0/pci8086,3410@9 [pcie3]  ENABLED
/pci@0,0/pci8086,342d  <pci.13,0>  OFFLINE
/pci@0,0/pci8086,342e  <pci.14,0>  OFFLINE
```

To show the binding driver names and instance numbers of the devices, enter the following command:

```
# hotplug list -d
pci@0,0  npe#0
    pci108e,4341  <pci.0,0>  OFFLINE  #0
```

例 1 Showing All Hotplug Connections *(続き)*

```
pci8086,3408  <pci.1,0>  OFFLINE  pcieb#0
pci8086,340a@3  <pci.3,0>  ONLINE  pcieb#1
    [Slot2]  EMPTY
pci8086,340c@5  <pci.5,0>  ONLINE  pcieb#2
    [Slot1]  ENABLED
    pci111d,8018@0  <pci.0,0>  ONLINE  pcieb#0
        pci111d,8018@2  <pci.2,0>  ONLINE  pcieb#6
            pci108e,f1bc@0  <pci.0,0>  ONLINE  e1000g#0
            pci108e,f1bc@0,1  <pci.0,1>  OFFLINE  e1000g#1
pci8086,340e@7  <pci.7,0>  ONLINE  pcieb#3
    [pcie4]  ENABLED
pci8086,3410@9  <pci.9,0>  ONLINE  pcieb#4
    [pcie3]  ENABLED
pci8086,342d  <pci.13,0>  OFFLINE  #0
pci8086,342e  <pci.14,0>  OFFLINE  #0
```

To show the status of all physical hotplug connectors. enter the following command:

```
# hotplug list -c
Connection             State         Description
-------------------------------------------------------
NEM0                   ENABLED       PCIe-Native
PCI-EM0                ENABLED       PCIe-Native
NEM1                   ENABLED       PCIe-Native
PCI-EM1                ENABLED       PCIe-Native
```

The table in the condition that multiple connectors with the same exist in the system:

```
# hotplug list -c
Connection   Note      State         Description
--------------------------------------------------------
NEM0                   ENABLED       PCIe-Native
PCI-EM0      *1        ENABLED       PCIe-Native
PCI-EM0      *2        ENABLED       PCIe-Native
NEM1                   ENABLED       PCIe-Native

Note:
Multiple connectors with the same name exist:

  [1] PCI-EM0   /pci@0,0/pci108e,4341@1a,2
  [2] PCI-EM0   /pci@1,0/pci108e,4341@1a,2
```

例 2 Reporting Failure During State Change Operation

If a change of state operation fails, an explanation is displayed to describe the failure. An attempt to offline a hotplug port with dependent devices that are currently in use by the system might fail as follows:

例 2 Reporting Failure During State Change Operation (続き)

```
# hotplug offline /pci@0,0/pci10de,5d@e pci.a,0
ERROR: devices or resources are busy.
pci108e,534a@a,0:
    { Network interface nge0 }
    { nge0: hosts IP addresses: 10.0.0.1 }
    { Plumbed IP Address }
```

例 3 Enabling User Interaction During State Change Operation

If there are multiple connectors with the same name in a system, the state change operations interact with the user to specify which connector needs to be operated upon.

```
# hotplug enable PCI-EM0
Multiple connectors with the same name exist:

   [1] PCI-EM0  /pci@0,0/pci108e,4341@1a,2
   [2] PCI-EM0  /pci@1,0/pci108e,4341@1a,2

Please select a connector, then press ENTER:
```

例 4 Displaying Bus-Specific Properties and Values

The following command displays all supported bus-specific properties and their possible values:

```
# hotplug get -o help /pci@0,0 pci.2,1
power_led=<on|off|blink>
fault_led=<on|off|blink>
active_led=<on|off|blink>
attn_led=<on|off|blink>
card_type=<type description>
board_type=<type description>
```

例 5 Displaying Bus-Specific Options

The following command displays the card type and the current state of the Power LED of a PCI hotplug connector:

```
# hotplug get -o card_type,power_led /pci@0,0 pci.2,1
card_type=fibre
power_led=on
```

例 6 Setting a Bus-Specific Property

The following command turns on the attention LED of a PCI hotplug connector:

```
# hotplug set -o attn_led=on /pci@0,0 pci.2,1
```

例 7 Installing Port Dependents

The following commands install dependent ports of an IOV physical function and then display the resulting IOV virtual functions that were probed.

```
# hotplug install /pci@400/pci@1/pci@0/pci@4 pci.0,1

# hotplug list -v /pci@400/pci@1/pci@0/pci@4 pci.0,1
<pci.0,1>  (ONLINE)
     { IOV physical function }
     { IOV virtual function 'pci.0,81' }
     { IOV virtual function 'pci.0,83' }
     { IOV virtual function 'pci.0,85' }
     { IOV virtual function 'pci.0,87' }
<pci.0,81>  (OFFLINE)
ethernet@0,81
<pci.0,83>  (OFFLINE)
ethernet@0,83
<pci.0,85>  (OFFLINE)
ethernet@0,85
<pci.0,87>  (OFFLINE)
ethernet@0,87
```

例 8 Uninstalling Port Dependents

The following command attempts to uninstall dependent ports of an IOV physical function, but fails when a dependent IOV virtual function is busy.

```
# hotplug uninstall /pci@400/pci@1/pci@0/pci@4 pci.0,0
ERROR: devices or resources are busy.
ethernet@0,81:
    { Network interface igvbf1 }
    { igvbf1: hosts IP addresses: 10.0.0.1 }
    { Plumbed IP Address }
```

終了ステータス

0
: Successful completion.

1
: Invalid command line options were specified.

2
: The specified path or connection does not exist.

3
: A fatal error occurred. One or more error messages are displayed on standard error.

4
: The hotplug service is not available.

5
: The connector name specified refers to multiple ports.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

cfgadm(1M), hotplugd(1M), getsubopt(3C), attributes(5)

診断

The following error message is displayed on systems that do not have any supported IO buses:

```
ERROR: there are no connections to display.
(See hotplug(1m) for more information.)
```

If this error message is seen, note that the system might still have other IO devices that support hotplugging, through the cfgadm(1M) command instead of hotplug.

注意事項

The hotplug service (FMRI svc:/system/hotplug) must be enabled as a prerequisite for using the hotplug command. The service is disabled by default. See hotplugd(1M).

The authorization solaris.hotplug.modify must be granted in order to perform change-of-state operations, or to install and uninstall dependent ports. Alternatively, the rights profile “Hotplug Management” can be granted, which includes that authorization.

Verbose usage information is gathered from the RCM framework. Its format and content is subject to change.

The following bus specific properties are supported in PCI bus controllers:

power_led | fault_led | attn_led | active_led
: States of a specific LED of a slot. The value could be on, off, or blink.

  They can all be used with get subcommand, but only property attn_led can be used with set subcommand.

card_type | board_type
: Type of a card or board of a slot.

  They can all be used with get subcommand, but neither can be used with set subcommand.

名前

hotplugd – hotplug daemon

形式

```
/usr/lib/hotplugd [-d]
```

機能説明

The hotplug daemon, `hotplugd`, provides user-level services for the management of hotplug connections. It is a system daemon started by the Service Management Facility (see `smf(5)`). Its fault management resource identifier (FMRI) is:

```
svc:/system/hotplug:default
```

Note that `hotplugd` is a Consolidation Private interface. See `attributes(5)`.

The `hotplug(1M)` command and any other client program that uses the private `libhotplug` library to query information about hotplug connections or initiate hotplug commands depends upon this daemon. The hotplug daemon is a door server which services requests from all `libhotplug` clients. The door interface is private.

Client applications use the private `libhotplug` interface to administer hotplug connections. `libhotplug` uses the door interface to administer hotplug connections through the hotplug daemon service. The hotplug daemon acts as a central location to serialize all hotplug operations and coordinate activities with all other parts of the system.

オプション

The following option is supported:

`-d`, `--debug`
: Run the daemon in standalone debug mode. Messages will be displayed on the controlling terminal instead of to syslog. And increased verbosity will be enabled to display more details about the internal operations of the daemon.

使用例

例 1 Enabling the Hotplug Service

The following command enables the hotplug service:

```
# svcadm enable svc:/system/hotplug:default
```

例 2 Disabling the Hotplug Service

The following command disables the hotplug service:

```
# svcadm disable svc:/system/hotplug:default
```

Errors

The hotplug daemon uses `syslog(3C)` to report status and error messages. All of the messages are logged with the `LOG_DAEMON` facility. Error messages are logged with the `LOG_ERR` and `LOG_NOTICE` priorities, and informational messages are logged with the `LOG_INFO` priority. The default entries in the `/etc/syslog.conf` file log all of the hotplug daemon error messages to the `/var/adm/messages` log.

ファイル

`/var/run/hotplugd_door`
: Hotplug daemon door

`/var/run/hotplugd_pid`
: Hotplug daemon lock file

`/usr/lib/hotplugd`
Hotplug daemon binary

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Consolidation Private

関連項目

`svcs(1)`, `hotplug(1M)`, `svcadm(1M)`, `syslog(3C)`, `syslog.conf(4)`, `attributes(5)`, `smf(5)`

注意事項

The `hotplugd` service is managed by the service management facility, `smf(5)`, under the fault management resource identifier (FMRI):

```
svc:/system/hotplug:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command. To administer the service, the authorization `solaris.smf.manage.hotplug` must be granted. Alternatively, the rights profile “Hotplug Management” can be granted.

The hotplug service must be enabled for the `hotplug(1M)` command and any other `libhotplug` client applications to function properly.

名前

htable – convert DoD Internet format host table

形式

`/usr/sbin/htable` *filename*

機能説明

`htable` converts a host table in the format specified by RFC 952 to the format used by the network library routines. Three files are created as a result of running `htable`: `hosts`, `networks`, and `gateways`. The `hosts` file is used by the `gethostbyname(3NSL)` routines in mapping host names to addresses. The `networks` file is used by the `getnetbyname(3SOCKET)` routines in mapping network names to numbers. The `gateways` file is used by the routing daemon to identify "passive" Internet gateways.

If any of the files `localhosts`, `localnetworks`, or `localgateways` are present in the current directory, the file's contents is prepended to the output file without interpretation. This allows sites to maintain local aliases and entries which are not normally present in the master database.

`htable` is best used in conjunction with the `gettable(1M)` program which retrieves the DoD Internet host table from a host.

ファイル

`localhosts`

`localnetworks`

`localgateways`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`gettable(1M)`, `gethostbyname(3NSL)`, `getnetbyname(3SOCKET)`, `attributes(5)`
Harrenstien, Ken, Mary Stahl, and Elizabeth Feinler, DoD Internet Host Table Specification, RFC 952, Network Information Center, SRI International, Menlo Park, California, October 1985.

注意事項

`htable` does not properly calculate the `gateways` file.

名前

ickey – install a client key for WAN boot

形式

```
/usr/lib/inet/wanboot/ickey [-d] [-o type=3des]

/usr/lib/inet/wanboot/ickey [-d] [-o type=aes]

/usr/lib/inet/wanboot/ickey [-d] [-o type=sha1]
```

機能説明

The `ickey` command is used to install WAN boot keys on a running UNIX system so that they can be used the next time the system is installed. You can store three different types of keys: 3DES and AES for encryption and an HMAC SHA-1 key for hashed verification.

`ickey` reads the key from standard input using `getpass(3C)` so that it does not appear on the command line. When installing keys on a remote system, you must take proper precautions to ensure that any keying materials are kept confidential. At a minimum, use `ssh(1)` to prevent interception of data in transit.

Keys are expected to be presented as strings of hexadecimal digits; they can (but need not) be preceded by a `0x` or `0X`.

The `ickey` command has a single option, described below. An argument of the type `-o type=`*keytype* is required.

オプション

The `ickey` command the following option.

`-d` Delete the key specified by the *keytype* argument.

終了ステータス

On success, `ickey` exits with status 0; if a problem occurs, a diagnostic message is printed and `ickey` exits with non-zero status.

ファイル

`/dev/openprom` WAN boot key storage driver

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Uncommitted

関連項目

`ssh(1)`, `openprom(7D)`, `attributes(5)`

名前

id – return user identity

形式

```
/usr/bin/id [-p] [user]
/usr/bin/id -a [-p] [user]
/usr/bin/id -G [-n] [user]
/usr/bin/id -g [-nr] [user]
/usr/bin/id -u [-nr] [user]
/usr/xpg4/bin/id [-p] [user]
/usr/xpg4/bin/id -a [-p] [user]
/usr/xpg4/bin/id -G [-n] [user]
/usr/xpg4/bin/id -g [-nr] [user]
/usr/xpg4/bin/id -u [-nr] [user]
```

機能説明

If no *user* operand is provided, the `id` utility writes the user and group IDs and the corresponding user and group names of the invoking process to standard output. If the effective and real IDs do not match, both are written. If multiple groups are supported by the underlying system, `/usr/xpg4/bin/id` also writes the supplementary group affiliations of the invoking process.

If a *user* operand is provided and the process has the appropriate privileges, the user and group IDs of the selected user are written. In this case, effective IDs are assumed to be identical to real IDs. If the selected user has more than one allowable group membership listed in the group database, `/usr/xpg4/bin/id` writes them in the same manner as the supplementary groups described in the preceding paragraph.

Formats

The following formats are used when the `LC_MESSAGES` locale category specifies the "C" locale. In other locales, the strings `uid`, `gid`, `euid`, `egid`, and `groups` may be replaced with more appropriate strings corresponding to the locale.

"uid=%u(%s) gid=%u(%s)\n" *<real user ID>*, *<user-name>*,
    <real group ID>, *<group-name>*

If the effective and real user IDs do not match, the following are inserted immediately before the `\n` character in the previous format:

" euid=%u(%s)"

with the following arguments added at the end of the argument list:

<effective user ID>, *<effective user-name>*

If the effective and real group IDs do not match, the following is inserted directly before the `\n` character in the format string (and after any addition resulting from the effective and real user IDs not matching):

" egid=%u(%s)"

with the following arguments added at the end of the argument list:

<effectivegroup-ID>, *<effectivegroupname>*

If the process has supplementary group affiliations or the selected user is allowed to belong to multiple groups, the first is added directly before the `NEWLINE` character in the format string:

`" groups=%u(%s)"`

with the following arguments added at the end of the argument list:

<supplementary group ID>, *<supplementary group name>*

and the necessary number of the following added after that for any remaining supplementary group IDs:

`",%u(%s)"`

and the necessary number of the following arguments added at the end of the argument list:

<supplementary group ID>, *<supplementary group name>*

If any of the user ID, group ID, effective user ID, effective group ID or supplementary/multiple group IDs cannot be mapped by the system into printable user or group names, the corresponding (`%s`) and name argument is omitted from the corresponding format string.

When any of the options are specified, the output format is as described under OPTIONS.

オプション

The following options are supported by both `/usr/bin/id` and `/usr/xpg4/bin/id`. The `-p` and `-a` options are invalid if specified with any of the `-G`, `-g`, or `-u` options.

`-p` Reports additionally the current project membership of the invoking process. The project is reported using the format:

`"projid=%u(%s)"`

which is inserted prior to the `\n` character of the default format described in the `Formats` section. The arguments

<project ID>,*<project name>*

are appended to the end of the argument list. If the project ID cannot be mapped by the system into a printable project name, the corresponding `(%s)` and name argument is omitted from the corresponding format string.

`-a` Reports user name, user ID and all the groups to which the user belongs.

`-G` Outputs all different group IDs (effective, real and supplementary) only, using the format `"%u\n"`. If there is more than one distinct group affiliation, output each such affiliation, using the format `" %u"`, before the `NEWLINE` character is output.

`-g` Outputs only the effective group ID, using the format `"%u\n"`.

`-n` Outputs the name in the format `"%s"` instead of the numeric ID using the format `"%u"`.

`-r` Outputs the real ID instead of the effective ID.

`-u` Outputs only the effective user ID, using the format `"%u\n"`.

オペランド

The following operand is supported:

user The user (login) name for which information is to be written.

環境変数

See `environ(5)` for descriptions of the following environment variables that affect the execution of `id`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

/usr/bin/id

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os, system/library/processor
Interface Stability	Committed

/usr/xpg4/bin/id

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Standard

関連項目

`fold(1)`, `logname(1)`, `who(1)`, `getgid(2)`, `getgroups(2)`, `getprojid(2)`, `getuid(2)`, `attributes(5)`, `environ(5)`, `standards(5)`

注意事項

Output produced by the `-G` option and by the default case could potentially produce very long lines on systems that support large numbers of supplementary groups.

名前

idmap – ネイティブアイデンティティーマッピングサービスの構成と管理

形式

```
idmap

idmap -f command-file

idmap add [-d] name1 name2

idmap dump [-n] [-v]

idmap export [-f file] format

idmap flush [-a]

idmap get-namemap name

idmap help

idmap import [-F] [-f file] format

idmap list

idmap remove [-t|-f] name

idmap remove -a

idmap remove [-d] name1 name2

idmap set-namemap [-a authenticationMethod] [-D bindDN]
     [-j passwdfile] name1 name2

idmap show [-c] [-v] [-V] identity [target-type]

idmap unset-namemap [-a authenticationMethod] [-D bindDN]
     [-j passwdfile] name [target-type]
```

機能説明

idmap ユーティリティーは、ネイティブアイデンティティーマッピングサービスの構成や管理に使用されます。

ネイティブアイデンティティーマッピングサービスでは、Windows のセキュリティー識別子 (SID) と POSIX のユーザー ID およびグループ ID (UID および GID) との次のタイプのマッピングをサポートします。

- 名前ベースマッピング。管理者が、Windows と UNIX のユーザーやグループのマッピングを名前に基づいて行います。
- 一時 **ID** マッピング。まだ名前に基づいてマッピングされていないすべての SID に、UID または GID が動的に割り当てられます。
- ローカル **SID** マッピング。一時的でない UID や GID が、アルゴリズムで生成されたローカル SID にマッピングされます。

idmap ユーティリティーを使用すると、名前ベースマッピングの作成や管理を行なったり、使用中のマッピングを監視したりできます。

idmap ユーティリティーの呼び出し時にサブコマンドもオプションも指定されなかった場合、ユーティリティーは標準入力からサブコマンドを読み取ります。標準入力が TTY であった場合、idmap コマンドは使用方法に関するメッセージを出力したあとに終了します。

マッピングメカニズム

idmapd(1M) デーモンは次のようにして、Windows のユーザーやグループの SID を UNIX の UID や GID にマッピングします。

1. SID が名前に基づいてマッピングされます。

   このマッピングでは、システム管理者が手動で設定した名前ベースマッピングが使用されます。

2. 名前ベースマッピングが見つからない場合、SID は動的に割り当てられた一時 ID にマッピングされます。

   この割り当てでは、2^{31} から 2^{32} - 2 の範囲内で次に使用可能な UID または GID が使用されます。

ローカル SID マッピングは、UNIX から Windows へのマッピングを行う場合に使用されます。

別名の問題が発生しないようにするには、すべてのファイルシステム、アーカイブおよびバックアップ形式、およびプロトコルが、SID を格納するか、あるいは 2^{31} から 2^{32} - 2 の範囲内の UID および GID をすべて nobody ユーザーおよびグループにマッピングする必要があります。

また、対角マッピングを作成することもできます。これは、Windows グループと Solaris ユーザーとの間、および Solaris グループと Windows ユーザーとの間のマッピングです。こうしたマッピングが必要となるのは、Windows があるグループアイデンティティーをファイル所有者として使用したり、その逆を行なったりする場合です。

名前ベースマッピング

名前ベースマッピングは、Windows のユーザーやグループと UNIX ネームサービスの対応する情報との間に、名前の等価性を確立します。これらのマッピングはリブート後も持続します。たとえば、次のコマンドは、Windows ユーザーを同じ名前の UNIX ユーザーにマッピングします。

```
# idmap add "winuser:*@mywindomain.com" "unixuser:*"
```

idmapd(1M) は、ディレクトリサービスを使用するように構成されている場合は、まず Active Directory (AD) またはネイティブ LDAP ディレクトリサービス、あるいはその両方のユーザーオブジェクトまたはグループオブジェクトに格納されたマッピング情報の使用を試みます。たとえば、特定の Windows ユーザーまたはグループの AD オブジェクトを拡張し、対応する Solaris ユーザーまたはグループの名前や数値 ID を含めることができます。同様に、特定の Solaris ユーザーまたはグループのネイティブ LDAP オブジェクトを拡張し、対応する Windows ユーザーまたはグループの名前を含めることができます。

AD またはネイティブ LDAP ディレクトリ、あるいはその両方に基づく名前マッピングを使用するように idmapd(1M) を構成するには、idmap サービスの対応するサービス管理機能 (SMF) プロパティーを設定します。詳細については、後述の「サービスプロパティー」を参照してください。

ディレクトリベースの名前マッピングが構成されていない場合や、構成されていても見つからない場合には、idmapd(1M) はローカルに格納された名前ベースのマッピング規則を処理します。

idmap では、Windows の既知の名前のマッピングがサポートされています。そのいくつかを次に一覧表示します。

```
Administrator
Guest
KRBTGT
Domain Admins
Domain Users
Domain Guest
Domain Computers
Domain Controllers
```

idmap 規則を追加すると、これらの既知の名前が標準形式に展開されます。つまり、デフォルトドメイン名が追加される (既知でない名前の場合) か、あるいは適切な組み込みのドメイン名が追加されます。このドメイン名は、その既知の名前の種類に応じて null、BUILTIN、またはローカルホスト名のいずれかになります。

次の一連の idmap コマンドは、既知でない名前 fred と既知の名前 administrator、guest がどのように処理されるかを示したものです。

```
# idmap add winname:fred unixuser:fredf
add     winname:fred    unixuser:fredf

# idmap add winname:administrator unixuser:root
add     winname:administrator   unixuser:root

# idmap add winname:guest unixuser:nobody
add     winname:guest   unixuser:nobody

# idmap add wingroup:administrators sysadmin
add     wingroup:administrators unixgroup:sysadmin

# idmap list
add     winname:Administrator@examplehost  unixuser:root
add     winname:Guest@examplehost  unixuser:nobody
add     wingroup:Administrators@BUILTIN unixgroup:sysadmin
add     winname:fred@example.com       unixuser:fredf
```

一時マッピング

idmapd デーモンは、その再起動の前後で一時 ID マッピングの維持を試みます。しかし、ID の維持に失敗した場合、デーモンは以前にマッピングされていた各 SID を新しい一時 UID または GID の値にマッピングします。このデーモンは、一時 UID または GID の再利用を行いません。idmapd デーモンが一時 UID および GID を使い果たしてしまうと、エラーが返され、さらに名前によるマッピングが不可能な SID のデフォルトの UID または GID が返されます。

動的な ID マッピングは、リブート後は維持されません。このため、UNIX の UID や GID に動的にマッピングされた SID はすべて、システムリブート後に別の ID にマッピングされる可能性が高くなります。

ローカル SID マッピング

名前ベースマッピングが見つからない場合、一時的でない UID や GID は、アルゴリズムで生成されたローカル SID にマッピングされます。マッピングの生成方法は次のとおりです。

```
local SID for UID = <machine SID> - <1000 + UID>
local SID for GID = <machine SID> - <2^31 + GID>
```

<machine SID> は、idmap サービスがその実行元となるホストに対して生成する一意の SID です。

規則検索順序

Windows 名から UNIX 名へのマッピング時には、名前ベースマッピング規則の検索が次の順序で実行されます。

1. *windows-name@domain* から ""
2. *windows-name@domain* から *unix-name*
3. *windows-name@** から ""
4. *windows-name@** から *unix-name*
5. **@domain* から *
6. **@domain* から ""
7. **@domain* から *unix-name*
8. **@** から *
9. **@** から ""
10. **@** から *unix-name*

UNIX 名から Windows 名へのマッピング時には、名前ベースマッピング規則の検索が次の順序で実行されます。

1. *unix-name* から ""
2. *unix-name* から *windows-name@domain*
3. * から **@domain*
4. * から ""
5. * から *windows-name@domain*

サービスプロパティー

idmapd(1M) デーモンの動作はサービスプロパティーによって決まります。これらのプロパティーは、SMF リポジトリ (smf(5) を参照) のプロパティーグループ config の下に格納されます。これらのプロパティーへのアクセスや変更を行うに

は、svccfg(1M) を使用しますが、それには solaris.smf.value.idmap 承認が必要となります。idmap サービスのサービスプロパティーを次に示します。

config/ad_unixuser_attr
: UNIX ユーザー名を含む AD 属性の名前を指定します。デフォルトはありません。

config/ad_unixgroup_attr
: UNIX グループ名を含む AD 属性の名前を指定します。デフォルトはありません。

config/nldap_winname_attr
: Windows ユーザー/グループ名を含むネイティブ LDAP 属性の名前を指定します。デフォルトはありません。

config/directory_based_mapping
: ディレクトリサービスに格納されたデータを使用したアイデンティティーマッピングのサポートを制御します。

  none を指定すると、ディレクトリベースのマッピングが無効になります。

  name を指定すると、前述のプロパティーを使用した名前ベースマッピングが有効になります。

  idmu を指定すると、Microsoft の Identity Management for UNIX (IDMU) を使用したマッピングが有効になります。この Windows コンポーネントを使用すると、管理者は各 Windows ユーザーの UNIX ユーザー ID を指定できるため、Windows アイデンティティーから対応する UNIX アイデンティティーへのマッピングが可能となります。使用されるのは、Solaris システムをメンバーに持つドメインからの IDMU データだけです。

サービスプロパティーを変更しても、実行中の idmap サービスには何の影響もありません。変更を反映するには、(svcadm(1M) を使用して) サービスを更新する必要があります。

オペランド

idmap コマンドで使用できるオペランドは、次のとおりです。

format
: export および import サブコマンド向けの、ユーザー名マッピングの記述形式を指定します。サポートされている外部形式は、Netapp usermap.cfg と Samba smbusers です。これらの外部形式を使用できるのはユーザーに対してのみであり、グループには使用できません。

- usermap.cfg 規則マッピング形式は次のとおりです。

  windows-username [*direction*] *unix-username*

  windows-username は、*domain**username* または *username*@*domain* のいずれかの形式の Windows ユーザー名です。

  unix-username は UNIX ユーザー名です。

direction は次のいずれかになります。

- == は双方向マッピングを意味します。これがデフォルトです。
- => または <= は単方向マッピングを意味します。

IP 修飾子はサポートされません。

- smbusers 規則マッピング形式は次のとおりです。

  unixname = *winname1* *winname2* ...

  winname にスペースが含まれる場合は、値を二重引用符で囲んでスペースをエスケープします。たとえば次のファイルは、idmap コマンドの有効な形式でスペースを指定する方法を示したものです。

```
$ cat myusermap
terry="Terry Maddox"
pat="Pat Flynn"
cal=cbrown
```

  これらのマッピングは、Windows 名から UNIX 名への単方向マッピングとしてインポートされます。

  この形式は、samba.org Web サイトで利用可能な smb.conf マニュアルページの「username map」エントリに基づいています。*windows-name* でのアスタリスク (*) の使用はサポートされています。ただし、@group 指令やマッピングの連鎖はサポートされていません。

  smbusers ファイルにマッピングエントリが 1 つも含まれていない場合、Samba はデフォルトで *windows-name* を、それと同等の *unix-name* が存在する場合はその名前にマッピングします。Samba と同じマッピングを設定する場合には、次の idmap コマンドを使用します。

```
idmap add -d "winuser:*@*" "unixuser:*"
```

identity

ユーザー名、ユーザー ID、グループ名、グループ ID のいずれかを指定します。*identity* は *type*:*value* として指定します。*type* は次のいずれかになります。

usid	テキスト形式の Windows ユーザー SID
gsid	テキスト形式の Windows グループ SID
sid	ユーザー、グループのいずれかに所属可能なテキスト形式の Windows グループ SID
uid	数値 POSIX UID
gid	数値 POSIX GID
unixuser	UNIX ユーザー名
unixgroup	UNIX グループ名

`winuser` Windows ユーザー名

`wingroup` Windows グループ名

`winname` Windows ユーザー名またはグループ名

value は、指定された *type* に適した数値または文字列です。たとえば、`unixgroup:staff` は、UNIX グループ名 `staff` を指定します。アイデンティティー `gid:10` は、UNIX グループ `staff` に対応する GID 10 を表します。

name
名前ベースマッピング規則で使用可能な UNIX 名 (`unixuser`、`unixgroup`) または Windows 名 (`winuser`、`wingroup`) を指定します。

Windows セキュリティーエンティティー名は、次のいずれかの方法で指定できます。

- *domain\name*
- *name@domain*
- *name*。デフォルトのマッピングドメインが使用されます。

name が空の文字列 ("") の場合、マッピングが抑制されます。マッピングされていない Windows ユーザーのログインを防止するため、"" を名前として使用しないようにしてください。

name でワイルドカード (*) が使用された場合、それは、ほかのマッピングに一致しないすべての名前に一致します。同様に、*name* がワイルドカードを使用した Windows 名 (`*@*`) である場合、それは、ほかのマッピングに一致しないすべてのドメイン内のすべての名前に一致します。

マッピング規則の両側で *name* にワイルドカードが使用された場合、Windows ユーザーと Solaris ユーザーの名前は同じになります。たとえば、規則が `"*@domain" == "*"` の場合、Windows ユーザー名の `jp@domain` はこの規則に一致し、Solaris ユーザー名の `jp` にマッピングされます。

name のタイプは、コマンド行に指定されたほかの引数やタイプから推定できる場合には、指定しなくてもかまいません。

target-type
`show` および `unset-namemap` サブコマンドで使用されます。`show` の場合、表示するべきマッピングタイプを指定します。たとえば、*target-type* が `sid` の場合、コマンド行に指定されたアイデンティティーにマッピングされている SID が `idmap show` から返されます。`unset-namemap` の場合、*name* オペランドで指定されたオブジェクト内の属性を特定します。

オプション

`idmap` コマンドでは 1 つのオプションと一連のサブコマンドがサポートされます。サブコマンドにもオプションがあります。

コマンド行のオプション

-f *command-file*

command-file から idmap のサブコマンドを読み取って実行します。idmap -f - コマンドは標準入力から読み取ります。このオプションは、サブコマンドでは使用されません。

サブコマンド

サポートされているサブコマンドは次のとおりです。

add [-d] *name1 name2*

名前ベースマッピング規則を追加します。名前マッピングはデフォルトでは双方向になります。-d オプションを使用すると、*name1* から *name2* への単方向マッピングが作成されます。

name1、*name2* の一方が Windows 名、他方が UNIX 名でなければなりません。Windows 名の場合、winname アイデンティティータイプは使用できません。代わりに、winuser または wingroup のいずれかのタイプを指定します。*name* オペランドについては、「オペランド」を参照してください。

マッピング対象となる名前が同じでマッピング方向が異なる 2 つの単方向マッピングは、1 つの双方向マッピングと同等です。

このサブコマンドでは solaris.admin.idmap.rules 承認が必要となります。

dump [-n] [-v]

前回のシステム起動以降にキャッシュされたすべてのマッピングをダンプします。-n オプションを指定すると、名前も表示されます。デフォルトで表示されるのは、sid、uid、および gid だけです。-v オプションを指定すると、マッピングがどのようにして生成されたかが表示されます。

export [-f *file*] *format*

名前ベースマッピング規則を、指定された *format* で標準出力にエクスポートします。-f *file* オプションを指定すると、指定された出力ファイルに規則が書き込まれます。

flush [-a]

将来のマッピング要求が現在の規則やディレクトリ情報に基づいて完全に処理されるように、アイデンティティーマッピングのキャッシュをフラッシュします。これは中断を伴わない操作です。規則が変更されると自動的にキャッシュがフラッシュされます。この手動操作を使用すると、新しく変更されたディレクトリ情報を強制的に有効化できます。

-a を指定すると、キャッシュがクリーンアップされます。この操作は、実行中の操作を中断する可能性があるため、アイドル状態のシステムでのみ使用するようにしてください。-a が必要になることは通常は考えられませんが、「白紙状態」のテストケースを設定する場合には、このオプションが役立つ可能性があります。

`get-namemap` *name*
指定された名前で表される AD またはネイティブ LDAP ユーザーオブジェクトまたはグループオブジェクトから、ディレクトリベース名前マッピングの情報を取得します。

`help`
使用方法に関するメッセージを表示します。

`import` [`-F`] [`-f` *file*] *format*
名前ベースマッピング規則を、指定された *format* を使用して標準入力からインポートします。`-f` *file* オプションを指定すると、指定されたファイルから規則が読み取られます。`-F` オプションを指定すると、既存の名前ベースマッピング規則がフラッシュされたあとで新しい規則が追加されます。

どの外部形式を使用する場合でも、インポートした規則を処理する際には、前述の「規則検索順序」節で説明したセマンティクスや順序が使用されます。

このサブコマンドでは `solaris.admin.idmap.rules` 承認が必要となります。

`list`
すべての名前ベースマッピング規則を一覧表示します。各規則は `idmap add` 形式で表示されます。

`remove` [`-t`|`-f`] *name*
指定された名前を含む名前ベースマッピング規則をすべて削除します。*name* には UNIX、Windows のいずれかのユーザー名またはグループ名を指定できます。

`-f` オプションを指定すると、*name* をマッピング元として使用する規則が削除されます。`-t` オプションを指定すると、*name* をマッピング先として使用する規則が削除されます。これらのオプションは相互に排他的です。

このサブコマンドでは `solaris.admin.idmap.rules` 承認が必要となります。

`remove -a`
すべての名前ベースマッピング規則を削除します。

このサブコマンドでは `solaris.admin.idmap.rules` 承認が必要となります。

`remove` [`-d`] *name1 name2*
name1 と *name2* との間の名前ベースマッピング規則を削除します。`-d` オプションを指定すると、*name1* から *name2* への規則が削除されます。

name1、*name2* の一方が Windows 名、他方が UNIX 名でなければなりません。

このサブコマンドでは `solaris.admin.idmap.rules` 承認が必要となります。

`set-namemap` [`-a` *authenticationMethod*] [`-D` *bindDN*] [`-j` *passwdfile*] *name1 name2*
AD またはネイティブ LDAP のユーザーオブジェクトまたはグループオブジェクトに、名前マッピング情報を設定します。*name1*、*name2* の一方が Windows 名、他方が UNIX 名でなければなりません。

name1 が Windows 名の場合は、*name1* で表される AD オブジェクトに UNIX 名 *name2* が追加されます。同様に、*name1* が UNIX 名の場合は、*name1* で表されるネイティブ LDAP エントリに Windows 名 *name2* が追加されます。

次のオプションがサポートされています。

`-a` *authenticationMethod*
ネイティブ LDAP エントリを変更するときの認証方式を指定します。詳細については、`ldapaddent(1M)` を参照してください。デフォルト値は `sasl/GSSAPI` です。

`-D` *bindDN*
識別名 *bindDN* を使用してディレクトリにバインドします。

`-j` *passwdfile*
ディレクトリへの認証用のパスワードを含むファイルを指定します。

`show` [`-c`] [`-v`] [`-V`] *name* [*target-type*]
指定された *name* のマッピング先となる *target-type* タイプのアイデンティティーを表示します。省略可能な *target-type* を省略した場合には、非対角マッピングが表示されます。

このサブコマンドがデフォルトで表示するのは、すでに確立されたマッピングだけです。`-c` オプションを指定すると、名前ベースマッピング構成の評価や ID の動的割り当てが強制的に行われます。

`-v` オプションを指定すると、このマッピングがどのように生成されたのかが表示されるほか、このマッピングが単純に生成されたのか、またはキャッシュから取得されたのかも表示されます。`-V` オプションを指定すると、拒否された暫定的な手順やアプローチなど、マッピングを決定する際に決定された具体的な手順が詳しく表示されます。

`unset-namemap` [`-a` *authenticationMethod*] [`-D` *bindDN*] [`-j` *passwdfile*] *name* [*target-type*]
指定された名前や省略可能なターゲットのタイプで表される AD またはネイティブ LDAP ユーザーオブジェクトまたはグループオブジェクトから、ディレクトリベース名前マッピングの情報を設定解除します。

オプションについては、`set-namemap` サブコマンドを参照してください。

使用例

例 1　名前ベースマッピング規則の両側でのワイルドカードの使用

次のコマンドは、`xyz.com` ドメインのすべての Windows ユーザー名を同じ名前の UNIX ユーザーにマッピングします。ただし、そのような UNIX ユーザーが存在しない場合や別のマッピングが存在する場合は除きます。そのような規則に一致したが UNIX ユーザー名が存在しなかった場合、一時 ID マッピングが使用されます。

```
# idmap add "winuser:*@xyz.com" "unixuser:*"
```

例 2　名前ベースマッピング規則の片側でのワイルドカードの使用

次のコマンドは、xyz.com ドメイン内のマッピングされていないすべての Windows ユーザーを guest UNIX ユーザーにマッピングします。-d オプションは、*@xyz.com ユーザーから guest ユーザーへの単方向マッピングを指定します。

```
# idmap add -d "winuser:*@xyz.com" unixuser:guest
```

例 3　双方向の名前ベースマッピング規則の追加

次のコマンドは、Windows ユーザー foobar@example.com から UNIX ユーザー foo へのマッピング、およびその逆のマッピングを行います。

```
# idmap add winuser:foobar@example.com unixuser:foo
```

次のコマンドは、前述のコマンドで追加されたマッピングの削除方法を示しています。

```
# idmap remove winuser:foobar@example.com unixuser:foo
```

例 4　UID - SID 間マッピングの表示

- 次のコマンドは、指定された UID uid:50000 のマッピング先となる SID を表示します。

  ```
  # idmap show uid:50000 sid
  uid:50000 -> usid:S-1-5-21-3223191800-2000
  ```

- 次のコマンドは、指定された Windows ユーザー名 joe@example.com のマッピング先となる UNIX ユーザー名を表示します。

  ```
  # idmap show joe@example.com unixuser
  winuser:joe@example.com -> unixuser:joes
  ```

例 5　キャッシュ内の SID - UID 間マッピングの一覧表示

次のコマンドは、キャッシュ内に存在するすべての SID- UID マッピングを表示します。

```
# idmap dump | grep "uid:"
usid:S-1-5-21-3223191800-2000    ==     uid:50000
usid:S-1-5-21-3223191800-2001    ==     uid:50001
usid:S-1-5-21-3223191800-2006    ==     uid:50010
usid:S-1-5-21-3223191900-3000    ==     uid:2147491840
usid:S-1-5-21-3223191700-4000    =>     uid:60001
```

例 6　idmap 要求のバッチ処理

次のコマンドは、idmap 要求のバッチ処理を行う方法を示したものです。この特定のコマンドシーケンスで実行される処理は、次のとおりです。

- foobar@example.com に対する以前の規則をすべて削除します。

例 6　idmap 要求のバッチ処理　　(続き)

- Windows ユーザー foobar@example.com から UNIX ユーザー bar へのマッピング、およびその逆のマッピングを行います。
- Windows グループ members から UNIX グループ staff へのマッピング、およびその逆のマッピングを行います。

```
# idmap <<EOF
       remove winuser:foobar@example.com
       add winuser:foobar@example.com unixuser:bar
       add wingroup:members unixgroup:staff
EOF
```

例 7　名前ベースマッピング規則の一覧表示

次のコマンドは、名前ベースマッピング規則を一覧表示する方法を示したものです。

```
# idmap list
add winuser:foobar@example.com unixuser:bar
add wingroup:members unixgroup:staff
```

例 8　usermap.cfg ファイルからの名前ベースマッピング規則のインポート

usermap.cfg ファイルを使用すると、名前ベースマッピング規則を構成できます。次の usermap.cfg ファイルは、Windows ユーザー foo@example.com を UNIX ユーザー foo にマッピングし、foobar@example.com を UNIX ユーザー foo にマッピングするマッピング規則を示しています。

```
# cat usermap.cfg
foo@example.com == foo
foobar@example.com => foo
```

次の idmap コマンドは、usermap.cfg の情報を idmapd データベースにインポートします。

```
# cat usermap.cfg | idmap import usermap.cfg
```

このコマンドは、1 つ前のコマンドと同じ処理を行います。

```
# idmap import -f usermap.cfg usermap.cfg
```

次のコマンドは、前述の idmap import コマンドと同等です。

```
# idmap <<EOF
       add winuser:foo@example.com unixuser:foo
       add -d winuser:foobar@example.com unixuser:foo
EOF
```

例 9 アイデンティティー機能マッピングや例外とともに使用する、名前ベースマッピングと一時 ID マッピング

次のコマンドは、example.com Windows ドメイン内のすべてのユーザーを同じ名前の UNIX ユーザーアカウントにマッピングします。また、このコマンドでは次の Windows ユーザーのマッピングも指定されていま
す。joe@example.com、jane.doe@example.com、administrator@example.com。すべてのドメイン内の administrator が nobody にマッピングされています。対応する UNIX アカウントを持たない Windows ユーザーはすべて、使用可能な一時 UID に動的にマッピングされます。

```
# idmap import usermap.cfg <<EOF
       joe@example.com == joes
       jane.doe@example.com == janed
       administrator@* => nobody
       *@example.com == *
       *@example.com => nobody
EOF
```

例 10 AD ユーザーオブジェクトへのディレクトリベース名前マッピングの追加

次のコマンドは、Windows ユーザー joe@example.com を UNIX ユーザー joe にマッピングするために、joe@example.com の AD オブジェクトにその UNIX 名を追加します。

```
# idmap set-namemap winuser:joe@example.com joes
```

例 11 ネイティブ LDAP ユーザーオブジェクトへのディレクトリベース名前マッピングの追加

次のコマンドは、UNIX ユーザー foo を Windows ユーザー foobar@example.com にマッピングするために、foo のネイティブ LDAP オブジェクトにその Windows 名を追加します。

```
# idmap set-namemap unixuser:foo foobar@example.com
```

例 12 AD ユーザーオブジェクトからのディレクトリベース名前マッピングの削除

次のコマンドは、joe@example.com を表す AD オブジェクトから UNIX ユーザー名 unixuser を削除します。

```
# idmap unset-namemap winuser:joe@example.com unixuser
```

終了ステータス

0 正常終了。

>0 エラーが発生しました。診断メッセージが標準エラー出力に書き込まれます。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	不確実

関連項目

svcs(1), idmapd(1M), ldapaddent(1M), svcadm(1M), svccfg(1M), ad(5), attributes(5), smf(5)

注意事項

idmapd サービスは、サービス管理機能 smf(5) によって管理されます。idmapd サービスのサービス識別子は、svc:/system/idmap です。

サービスの有効化、無効化、再起動などの管理操作をこのサービスで実行するには、svcadm コマンドを使用します。これらのアクションには solaris.smf.manage.idmap 承認が必要となります。サービスのステータスに対するクエリーを実行するには、svcs コマンドを使用します。

Windows ユーザー名では大文字と小文字の区別がありませんが、UNIX ユーザー名ではその区別があります。idmap の名前規則や idmap show コマンド行で Windows 名が大文字、小文字のどちらで表示されるかは、重要ではありません。

すべて小文字のユーザー名を使用するのが UNIX 環境の一般的な方法であるため、ワイルドカード名前規則による Windows 名から UNIX ユーザー/グループ名へのマッピングは、次のようになります。まず、標準の Windows 名 (つまり、ディレクトリに表示されるとおりの大文字、小文字の使い方を採用した名前) が、UNIX ユーザー名またはグループ名として使用されます。そのような UNIX エンティティーが存在しない場合には、Windows 名の大文字がすべて小文字に変換され、その結果が UNIX ユーザー名またはグループ名として使用されます。

この大文字、小文字に応じた処理が異なるため、一致すると思われるユーザー名が一致として認識されない場合が発生します。大文字、小文字の使い方だけが異なるそのような文字列のペアを処理するための規則を作成する必要があります。たとえば、Windows ユーザー sam@example を Solaris ユーザー Sam にマッピングするには、次の規則を作成する必要があります。

```
# idmap add "winuser:*@example" "unixuser:*"
# idmap add winuser:sam@example unixuser:Sam
```

Active Directory スキーマの変更に関するガイダンスについては、Microsoft のドキュメント『スキーマおよび表示 Specifier を使った Active Directory の高度な管理』を参照してください。このドキュメントは、technet Web サイト http://technet.microsoft.com/ja-jp/library/bb727064.aspx から入手できます。

名前

idmapd – ネイティブアイデンティティーマッピングサービスデーモン

形式

/usr/lib/idmapd

機能説明

idmapd デーモンは、Windows セキュリティー識別子 (SID) から POSIX 識別子 (UID/GID) へのマッピングおよびその逆を行います。

idmap(1M) ユーティリティーは、idmapd デーモンへのフロントエンドを提供します。

ファイル

/var/idmap/idmap.db
: ローカルの名前ベース ID マッピング規則の格納先となるデータベース。このデータベースの内容は非公開です。このデータベースへのアクセスや変更を直接行わないようにしてください。

/var/run/idmap/idmap.db
: 一時 ID マッピングおよび名前ベースマッピングによって生成された ID マッピングのキャッシュ先となるデータベース。このデータベースの内容は非公開です。このデータベースへのアクセスや変更を直接行わないようにしてください。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照。

これらのコンポーネントのインタフェースの安定性は、次のとおりです。

svc:/system/idmap
: 確実

/var/idmap/idmap.db
: プロジェクト非公開

/var/run/idmap/idmap.db
: プロジェクト非公開

関連項目

svcs(1), idmap(1M), svcadm(1M), svccfg(1M), defaultdomain(4), attributes(5), smf(5)

注意事項

idmapd サービスは、サービス管理機能 (SMF) によって管理されます。idmapd サービスのサービス識別子は、svc:/system/idmap です。

サービスの有効化、無効化、再起動などの管理操作をこのサービスで実行するには、svcadm コマンドを使用します。これらのアクションには solaris.smf.manage.idmap 承認が必要となります。サービスのステータスに対するクエリーを実行するには、svcs コマンドを使用します。

このデーモンの機能は、Solaris オペレーティングシステムの将来のリリースで変更される可能性があります。

名前

idsconfig – prepare a Directory Server Enterprise Edition (DSEE) to be populated with data and serve LDAP clients

形式

```
/usr/lib/ldap/idsconfig [-v] [-i input_configfile]
     [-o output_configfile]
```

機能説明

Use the `idsconfig` tool to set up a Directory Server Enterprise Edition (DSEE). You can specify the input configuration file with the `-i` option on the command line. Alternatively, the tool will prompt the user for configuration information. The input configuration file is created by `idsconfig` with the `-o` option on a previous run.

The first time a server is set up, the user is prompted for all the required information. Future installations on that machine can use the configuration file previously generated by `idsconfig` using the `-o` option.

The output configuration file contains the directory administrator's password in clear text. Thus, if you are creating an output configuration file, take appropriate security precautions.

You should back up the directory server's configuration and data prior to running this command.

オプション

The following options are supported:

`-i` *input_configfile*
: Specify the file name for `idsconfig` to use as a configuration file. This file will be read by `idsconfig`, and the values in the file will be used to configure the server. Do not manually edit *input_configfile*. The *input_configfile* is only partially validated, as `idsconfig` assumes that the file was created by a previous invocation of the command.

`-o` *output_configfile*
: Create a configuration file.

`-v`
: Verbose output.

オペランド

The following operands are supported:

input_configfile
: Name of configuration file for `idsconfig` to use.

output_configfile
: Configuration file created by `idsconfig`.

使用例

例 1　Prompting the User for Input

In the following example, the user is prompted for information to set up DSEE.

```
example# idsconfig
```

例 2 Creating an Output Configuration File

In the following example, the user is prompted for information to set up DSEE, and an output configuration file, `config.1`, is created when completed.

```
example# idsconfig -o config.1
```

例 3 Setting up DSEE Using the Specified Configuration File

In the following example, DSEE is set up by using the values specified in the configuration file, `config.1`. The verbose mode is specified, so detailed information will print to the screen.

```
example# idsconfig -v -i config.1
```

終了ステータス

The following exit values are returned:

0
: Successful completion.

>0
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis
Interface Stability	Committed

関連項目

`ldapadd(1)`, `ldapdelete(1)`, `ldaplist(1)`, `ldapmodify(1)`, `ldapmodrdn(1)`, `ldapsearch(1)`, `ldap_cachemgr(1M)`, `ldapaddent(1M)`, `ldapclient(1M)`, `resolv.conf(4)`, `attributes(5)`,`ldap(5)`

名前

ifconfig – ネットワークインタフェースのパラメータの構成

形式

```
ifconfig interface [address_family] [address [/prefix_length]
 [dest_address]] [addif address [/prefix_length]]
 [removeif address [/prefix_length]] [arp | -arp]
 [auth_algs authentication algorithm] [encr_algs encryption algorithm]
 [encr_auth_algs authentication algorithm] [auto-revarp]
 [broadcast address] [deprecated | -deprecated]
 [preferred | -preferred] [destination dest_address]
 [ether [address]] [failover | -failover] [group
 [name | ""]] [index if_index] [ipmp] [metric n] [modlist]
 [modinsert mod_name@pos] [modremove mod_name@pos]
 [mtu n] [netmask mask] [plumb] [unplumb] [private
 | -private] [nud | -nud] [set [address] [/netmask]]
 [standby | -standby] [subnet subnet_address] [tdst
 tunnel_dest_address] [token address/prefix_length]
 [tsrc tunnel_src_address] [trailers | -trailers]
 [up] [down] [usesrc [name | none]] [xmit | -xmit]
 [encaplimit n | -encaplimit] [thoplimit n] [router
 | -router] [zone zonename | -zone | -all-zones]

ifconfig [address_family] interface {auto-dhcp | dhcp} [primary]
 [wait seconds] drop | extend | inform | ping
 | release | start | status
```

機能説明

コマンド `ifconfig` は、ネットワークインタフェースにアドレスを割り当てたり、ネットワークインタフェースのパラメータを構成したりするために使用されます。`ifconfig` コマンドで構成されたネットワークインタフェースは、リブート後は無効になります。ネットワークインタフェースの永続的な構成を行うには、`ipadm(1M)` コマンドを使用する必要があります。オプションが 1 つも指定されなかった場合、`ifconfig` は、ネットワークインタフェースの現在の構成を表示します。アドレスファミリが指定された場合、`ifconfig` は、そのアドレスファミリに固有の詳細のみを報告します。ネットワークインタフェースの構成を変更できるのは、特権ユーザーだけです。中括弧 (`{ }`) で囲まれたオプションは、そのうちのどれか 1 つのオプションを指定することを示します。

DHCP 構成

`auto-dhcp` または `dhcp` 引数を使用する `ifconfig` の形式は、インタフェースの動的ホスト構成プロトコル (「DHCP」) 構成を制御するために使用されます。このモードでは、DHCP クライアントデーモンである `dhcpagent(1M)` の動作を制御する目的で、`ifconfig` が使用されます。あるインタフェースがいったん `start` オペランド経由で DHCP の制御下に置かれたら、通常の操作では、そのインタフェースのアドレスや特性を `ifconfig` を使って変更するべきではありません。DHCP 配下の特定のインタフェースのアドレスが変更されると、`dhcpagent` はそのインタフェースを制御対象外にします。

オプション

次のオプションがサポートされています。

`addif` *address*
指定された物理インタフェース上に、次の未使用の論理インタフェースを作成します。

`all-zones`
インタフェースをシステム上のすべての共有 IP ゾーンで使用できるようにします。データを提供する対象のゾーンは、`tnzonecfg` データベースを使用して決定されます。このオプションを使用できるのは、システムが Solaris Trusted Extensions で構成されている場合だけです。

`tnzonecfg` データベースについては、『Solaris Trusted Extensions リファレンスマニュアル』に含まれている `tnzonecfg(4)` のマニュアルページを参照してください。

`anycast`
`ANYCAST` フラグを設定して、論理インタフェースをエニーキャストアドレスとしてマークします。エニーキャストの詳細については、下記の「インタフェースフラグ」を参照してください。

`-anycast`
`ANYCAST` フラグをクリアして、論理インタフェースを非エニーキャストアドレスとしてマークします。

`arp`
ネットワークレベルのアドレスとリンクレベルのアドレスとの間のマッピングを行う際に、アドレス解決プロトコル (「ARP」) の使用を有効にします (デフォルト)。これは現時点では、IPv4 アドレスと MAC アドレス間のマッピング用として実装されています。

`-arp`
物理インタフェースでの ARP の使用を無効にします。IPMP IP インタフェースでは ARP を無効にできません。

`auth_algs` *authentication algorithm*
特定のトンネルに対し、指定された認証アルゴリズムを使って IPsec AH を有効にします。このアルゴリズムは、数字、アルゴリズム名のどちらで指定してもかまいません。どのアルゴリズムでもかまわないことを示す *any* も使用できます。すべての IPsec トンネルプロパティーは、同一コマンド行に指定する必要があります。トンネルのセキュリティーを無効にするには、`auth_alg` に `none` を指定します。

トンネルのセキュリティープロパティーを設定する際、現在では `ipsecconf(1M)` を使用する方が適切です。`ipsecconf` を使用してトンネルのセキュリティープロパティーが設定されている場合、このキーワードはトンネルに影響を与えません。

`auto-dhcp`
このインタフェースのアドレスを、DHCP を使って自動的に取得します。このオプションには、`dhcp` という名前の、完全に等価な別名があります。

IPv6 の場合、指定するインタフェースは、リンクローカルアドレスが設定されている 0 番目の論理インタフェース (物理インタフェース名) でなければなりません。

primary
このインタフェースを `primary` として定義します。このインタフェースは、クライアント全体の構成データを配信するための優先インタフェースとして定義されます。主インタフェースになれるインタフェースは、一度に 1 つだけです。その後、別のインタフェースが主インタフェースとして選択された場合、以前の主インタフェースはそれに置き換えられます。クライアントワークステーションのブート完了後に特定のインタフェースを主インタフェースとして指定することは、あまり意味がありません。これは、多くのアプリケーションはすでにブートされており、以前の主インタフェースから読み取ったデータに基づいて構成されているからです。

wait *seconds*
`ifconfig` コマンドは、操作が完了するか、指定された時間が経過するまで待機します。実際の待機時間はどちらか早いほうになります。待機時間が指定されず、かつ操作がすぐに完了できない性質のものであった場合、`ifconfig` は 30 秒間、要求された処理が完了するのを待ちます。シンボリック値 `forever` も、リテラルな意味で使用できます。

drop
DHCP サーバーには通知せずに、指定されたインタフェースを DHCP の制御対象外にします。また、あとで使用できるように現在のリースを記録します。さらに IPv4 の場合は、IP アドレスをゼロに設定します。IPv6 の場合は、`dhcpagent` によって plumb されたすべての論理インタフェースを unplumb します。

extend
インタフェースの IP アドレスのリースを延長しようとします。これは必須ではありません。リースが期限切れになる前に、エージェントによって自動的にリースが延長されるからです。

inform
DHCP からネットワーク構成パラメータを取得します。その際、IP アドレスのリースは取得しません。これは、DHCP 以外のメカニズムを使って IP アドレスを取得する場合に役立ちます。

ping
指定されたインタフェースが DHCP の制御下にあるかどうか、つまり、そのインタフェースが DHCP エージェントによって管理されており、かつ正しく動作しているかをチェックします。終了ステータス `0` は、成功したことを意味します。

release
サーバーに通知してインタフェースの IP アドレスを解放し、現在のリースを破棄します。IPv4 の場合は、IP アドレスをゼロに設定します。IPv6 の場合は、`dhcpagent` によって plumb されたすべての論理インタフェースを unplumb します。

start
インタフェース上で DHCP を開始します。

status
インタフェースの DHCP 構成ステータスを表示します。

`auto-revarp`
逆アドレス解決プロトコル (「RARP」) を使ってこのインタフェースのアドレスを自動的に取得します。IPoIB (IP over InfiniBand) など、RARP をサポートしていないインタフェースおよび IPv6 インタフェースでは、この処理は失敗します。

`broadcast` *address*
IPv4 専用。ネットワークへのブロードキャストを示すために使用するアドレスを指定します。デフォルトのブロードキャストアドレスは、ホスト部分がすべて 1 であるようなアドレスです。ブロードキャストの値として「+」(プラス記号) を指定した場合、ブロードキャストアドレスは、(新しくなった可能性のある) アドレスとネットマスクに適したデフォルトにリセットされます。`ifconfig` の引数は左から右に解釈されます。したがって、

```
example% ifconfig -a netmask + broadcast +
```

および

```
example% ifconfig -a broadcast + netmask +
```

では、インタフェースのブロードキャストアドレスに異なる値が割り当てられる可能性があります。

`deprecated`
論理インタフェースを非推奨としてマークします。非推奨のインタフェースに関連付けられたアドレスが送信パケットの発信元アドレスとして使用されるのは、そのインタフェース上で利用可能なアドレスがほかに存在しない場合と、アプリケーションがそのアドレスに明示的にバインドした場合だけです。ステータス表示では、`DEPRECATED` がフラグの一部として表示されます。`ifconfig` でサポートされるフラグについては、インタフェースフラグ を参照してください。

`-deprecated`
論理インタフェースを「非推奨でない」としてマークします。そのようなインタフェースに関連付けられたアドレスは、送信パケットの発信元アドレスとして使用できます。

`preferred`
論理インタフェースを優先としてマークします。このオプションは IPv6 アドレスでのみ有効です。優先論理インタフェースに割り当てられたアドレスは、シス

テム上で構成されたほかのどのアドレスよりも、発信元アドレスとして優先されます。ただし、そのアドレスが着信先アドレスから見て不適切なスコープを持つ場合はその限りではありません。優先アドレスは、それらがどの物理インタフェースに割り当てられているかにかかわらず、常に発信元アドレスとして使用されます。たとえば、ループバックインタフェース上で優先発信元アドレスを構成し、そのアドレスの到達可能性を、経路指定プロトコルを使って通知できます。

`-preferred`
論理インタフェースを非優先としてマークします。

`destination` *dest_address*
ポイントツーポイントインタフェースの着信先アドレスを設定します。

`dhcp`
このオプションは、オプション `auto-dhcp` の別名です。

`down`
論理インタフェースを「down」としてマークします。(つまり、`IFF_UP` ビットをオフにします)。特定の論理インタフェースが「down」としてマークされると、システムは、そのインタフェースに割り当てられたアドレスを送信パケットの発信元アドレスとして使用しようとしなくなるほか、そのアドレス宛の受信パケットをこのホスト宛のものとして認識しなくなります。さらに、ある物理インタフェース上のすべての論理インタフェースが「down」になった場合、その物理インタフェースそのものが無効になります。

特定の論理インタフェースが停止すると、`route(1M)` コマンドの `-ifp` オプションまたは `route(7P)` ソケットの `RTA_IFP` を使ってそのインタフェースを出力として指定した経路がすべて、転送テーブルから削除されます。`RTF_STATIC` とマークされた経路はインタフェース復旧時にテーブルに戻されますが、`RTF_STATIC` とマークされていない経路は単に削除されます。

特定のゲートウェイアドレスに到達するために使用可能な論理インタフェースのすべてが停止した場合 (直前の段落で説明したインタフェースオプションなしで指定された場合)、その影響を受けるゲートウェイ経路は、`RTF_BLACKHOLE` フラグが設定されている場合と同様に処理されます。一致するすべてのパケットは、そのゲートウェイに到達できずに破棄されます。

`encaplimit` *n*
インタフェースのトンネルカプセル化制限を n に設定します。このオプションは、IPv6 内 IPv4 トンネルと IPv6 内 IPv6 トンネルにしか適用されず、単にベースとなる IPv6 トンネルリンクの `encaplimit` リンクプロパティーを変更するだけです (`dladm(1M)` を参照)。トンネルカプセル化制限は、特定のパケットが任意のトンネルを出る前にさらにいくつのトンネルに入れるか (つまりトンネルのネストレベル) を制御します。

このオプションは廃止され、`dladm(1M)` の `encaplimit` リンクプロパティーで置き換えられました。

`-encaplimit`
トンネルカプセル化制限の生成を無効にします。このオプションを使用できるのは、IPv6 内 IPv4 トンネルと IPv6 内 IPv6 トンネルに対してのみです。これは単に、ベースとなる IPv6 トンネルリンクの `encaplimit` リンクプロパティーを 0 に設定するだけです (`dladm(1M)` `encaplimit` を参照)。

このオプションは廃止され、`dladm(1M)` の `encaplimit` リンクプロパティーで置き換えられました。

`encr_auth_algs` *authentication algorithm*
特定のトンネルに対し、指定された認証アルゴリズムを使って IPsec ESP を有効にします。これは、数字、アルゴリズム名のどちらで指定してもかまいません。どのアルゴリズムでもかまわないことを示す `any` や `none` も使用できます。ESP 暗号化アルゴリズムは指定されたがその認証アルゴリズムが指定されなかった場合、ESP 認証アルゴリズムのデフォルト値は `any` になります。

トンネルのセキュリティープロパティーを設定する際、現在では `ipsecconf(1M)` を使用する方が適切です。`ipsecconf` を使用してトンネルのセキュリティープロパティーが設定されている場合、このキーワードはトンネルに影響を与えません。

`encr_algs` *encryption algorithm*
特定のトンネルに対し、指定された暗号化アルゴリズムを使って IPsec ESP を有効にします。これは、数字、アルゴリズム名のどちらで指定してもかまいません。すべての IPsec トンネルプロパティーは同一コマンド行に指定する必要があるので、注意してください。トンネルセキュリティーを無効にするには、`encr_alg` の値として `none` を指定します。ESP 認証アルゴリズムは指定されたがその暗号化アルゴリズムが指定されなかった場合、ESP 暗号化アルゴリズムのデフォルト値は `null` になります。

トンネルのセキュリティープロパティーを設定する際、現在では `ipsecconf(1M)` を使用する方が適切です。`ipsecconf` を使用してトンネルのセキュリティープロパティーが設定されている場合、このキーワードはトンネルに影響を与えません。

`ether` [*address*]
アドレスが指定されなかった場合、ユーザーがルートであるか、あるいは対象データリンクをオープンできるだけの十分な特権を備えていれば、現在の Ethernet アドレス情報を表示します。

それ以外の場合、ユーザーがルートであるか十分な特権を備えていれば、インタフェースの Ethernet アドレスを *address* に設定します。このアドレスは Ethernet アドレスであり、*x:x:x:x:x:x* として表現されます。ここで、*x* は 0 から FF までの 16 進数です。同様に、IPoIB (IP over InfiniBand) インタフェースの場合、このアドレスは、`0` から `FF` までの 16 進数がコロンで区切られた 20 バイトの文字列になります。

Ethernet インタフェースカードの中には、固有のアドレスを持つものもあります (すべてではない)。固有のアドレスを持たないカードを使用する場合には、IEEE 802.3 仕様の 3.2.3(4) セクションを参照し、ローカル管理アドレス空間の定義を確認してください。IPMP グローバル内の IP インタフェースは必ず、一意のハードウェアアドレスを持つ必要があります。in.mpathd(1M) を参照してください。

-failover
論理インタフェースに NOFAILOVER を設定します。これにより、in.mpathd が関連する物理 IP インタフェースに対してプローブベースの障害検出を実行する際に、関連するアドレスが使用可能となります。副作用として、DEPRECATED も論理インタフェースに設定されます。IPMP IP インタフェースでは、この操作を行うことはできません。

failover
論理インタフェースの NOFAILOVER をクリアします。これはデフォルトです。これらの論理インタフェースは、開始時に移行される可能性があります (「IP マルチパスグループ」を参照)。

group [*name* |""]
これを物理インタフェースに適用した場合、そのインタフェースが指定されたグループ内に配置されます。そのグループが存在しない場合は、1 つ以上の IPMP IP インタフェース (IPv4 用、IPv6 用、またはその両方) とともに作成されます。NOFAILOVER とマークされていない UP アドレスはすべて、IPMP IP インタフェースに移行される可能性があります (「IP マルチパスグループ」を参照)。グループ名として "" を指定すると、物理 IP インタフェースがグループから削除されます。

これを物理 IPMP IP インタフェースに適用した場合、IPMP グループの名前が変更されて新しい名前になります。その名前がすでに存在しているか、名前として "" が指定された場合は、処理が失敗します。IPMP グループの名前を変更することはお勧めできません。代わりに、ipmp サブコマンドを使用して IPMP IP インタフェースを作成する際に意味のある名前を指定するべきです。システムでは、この名前が IPMP グループ名としても使用されます。

index *n*
インタフェースのインタフェースインデックスを変更します。*n* の値は、ほかのインタフェースで使用されていないインタフェースインデックス (*if_index*) でなければなりません。*if_index* は 0 でない正数であり、システム上のネットワークインタフェースを一意に識別します。

ipmp
指定された名前の IPMP IP インタフェースを作成します。インタフェースの作成は、IPv4 用と IPv6 用とで別々に行う必要があります。コマンドが IPv4、IPv6 のどちらに適用されるかは、*address_family* パラメータによって決まります (指定されていない場合は IPv4 に適用される)。IPMP IP インタフェースには必ず IPMP フラグが設定されます。

`metric` *n*
インタフェースの経路指定メトリックを *n* に設定します。値が指定されなかった場合のデフォルトは、`0` になります。経路指定メトリックは、経路指定プロトコルによって使用されます。メトリックが高いほど、その経路は好まれません。メトリックは、着信先となるネットワークまたはホストへの加算ホップとしてカウントされます。

`modinsert` *mod_name@pos*
mod_name という名前のモジュールを、デバイスのストリーム内の位置 *pos* に挿入します。この位置は、ストリームの先頭からの相対位置です。位置 `0` は、ストリームの先頭のすぐ下を意味します。

`modlist` オプションの例に基づいて次のコマンドを使用すると、`ipqos` という名前のモジュールが、`ip` モジュールの下、firewall モジュールの上に挿入されます。

```
example% ifconfig eri0 modinsert ipqos@2
```

このあと、デバイスのストリーム内のすべてのモジュールを一覧表示した結果を、次に示します。

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 ipqos
3 firewall
4 eri
```

`modlist`
デバイスのストリーム内のすべてのモジュールを一覧表示します。

次の例では、デバイスのストリーム内のすべてのモジュールを一覧表示しています。

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 firewall
4 eri
```

`modremove` *mod_name@pos*
mod_name という名前のモジュールを、デバイスのストリーム内の位置 *pos* から削除します。この位置は、ストリームの先頭からの相対位置です。

`modinsert` オプションの例に基づいて次のコマンドを使用すると、`ipqos` モジュール挿入後のストリームから firewall モジュールが削除されます。

```
example% ifconfig eri0 modremove firewall@3
```

このあと、デバイスのストリーム内のすべてのモジュールを一覧表示した結果を、次に示します。

```
example% ifconfig eri0 modlist
0 arp
1 ip
2 ipqos
3 eri
```

ip や tun などのコア IP スタックモジュールは削除できません。

mtu *n*

インタフェースの最大転送単位を *n* に設定します。多くのネットワークタイプでは、mtu には上限があります。たとえば、Ethernet の場合は 1500 です。このオプションを使用すると、対象のインタフェース上に FIXEDMTU フラグが設定されます。

netmask *mask*

IPv4 専用。ネットワークをサブネットワークに分割する際にアドレスの何ビットを確保するかを指定します。マスクにはローカルアドレスのネットワーク部とサブネット部が含まれます。これは、アドレスのホストフィールドから取得されます。マスクでは、32 ビットアドレス内でネットワーク部とサブネット部用として使用するべきビット位置に 1 が、ホスト部に対するビット位置に 0 が、それぞれ含まれています。マスクには少なくとも標準ネットワーク部を含めてください。また、サブネットフィールドはネットワーク部に隣接するようにしてください。マスクの指定方法には次の 4 つがあります。

1. 0x で始まる単一の 16 進数の使用
2. ドット表記アドレスの使用
3. 「+」(プラス記号) アドレスの使用
4. ネットワークデータベース networks(4) 内に収められた擬似ホスト名/擬似ネットワーク名の使用

ネットマスク値として「+」(プラス記号) が指定された場合、netmasks(4) データベース内でマスクの検索が行われます。この検索では、データベース内で一致する最長のネットマスクを見つけるために、インタフェースの IPv4 アドレスをキーとして検索を開始し、アドレスのより多くの下位ビットを順次マスキングしていきます。この反復的な検索手法により、1 つのネットワーク番号内でさまざまな長さのサブネットマスクが使用されている場合に netmasks(4) データベースをネットマスク指定用として使用できることが保証されます。

擬似ホスト名/擬似ネットワーク名がネットマスク値として指定された場合、hosts または networks データベース内でネットマスクデータが検索されます。まず、gethostbyname(3NSL)を使って名前の検索が行われます。名前がそこに見つからなかった場合には、getnetbyname(3SOCKET) で名前の検索が行われます。これらのインタフェースは通常、実際の値を取得するのにどのデータストア (複数可) を使用するべきかを、nsswitch.conf(4) を使って判定します。

inet と inet6 のどちらでも、*mask* が表現しているのと同じ情報を、*address* パラメータに付随する *prefix_length* として指定することができます。

`nud`
特定のポイントツーポイント物理インタフェース上で近傍不到達検出メカニズムを有効にします。

`-nud`
特定のポイントツーポイント物理インタフェース上で近傍不到達検出メカニズムを無効にします。

`plumb`
物理 IP インタフェースの場合、その物理インタフェース名に関連付けられたデータリンクを開き、IP がデータリンクを使用するために必要な plumb の設定を行います。このコマンドを論理インタフェース名とともに使用した場合、その名前を持つ論理インタフェースが既存の物理 IP インタフェース上に作成されます。

インタフェースは、*address_family* パラメータに従って IPv4 用および IPv6 用 (指定されていない場合は IPv4 用) に個別に plumb する必要があります。plumb がまだ完了していないインタフェースは、`ifconfig -a` で表示されません。

IPMP IP インタフェースは特定のデータリンクに関連付けられないため、代わりに `ipmp` サブコマンドで作成されます。

`private`
指定された論理インタフェースは通知するべきでないことを、`in.routed` 経路指定デーモンに伝えます。

`-private`
通知されないインタフェースを指定します。

`removeif` *address*
指定された物理インタフェース上の論理インタフェースのうち、指定された *address* に一致するものを削除します。

`router`
インタフェース上の IP 転送を有効にします。有効にした場合、そのインタフェースは `ROUTER` としてマークされ、そのインタフェースへの、およびそのインタフェースからの、IP パケットの転送が行えるようになります。IPMP グループ内のいずれかの IP インタフェースで `ROUTER` を有効にすると、その IPMP グループ内のすべての IP インタフェースでそれが有効になります。

`-router`
インタフェース上の IP 転送を無効にします。そのインタフェースへの、およびそのインタフェースからの、IP パケットの転送は行われません。IPMP グループ内のいずれかの IP インタフェースで `ROUTER` を無効にすると、その IPMP グループ内のすべての IP インタフェースでそれが無効になります。

`set`
特定の論理インタフェースの *address* または *prefix_length*、あるいはその両方を設定します。

`standby`
物理 IP インタフェースを STANDBY インタフェースとしてマークします。IPMP グループの一部となっているインタフェースを STANDBY とマークした場合、IPMP グループ内の別のインタフェースが使用不可能にならないかぎり、そのインタフェースがデータトラフィックに使用されることはありません。正常に機能している STANDBY インタフェースがデータトラフィックに使用されていない場合、そのインタフェースはさらに INACTIVE とマークされます。IPMP IP インタフェースでは、この操作を行うことはできません。

`-standby`
インタフェースの STANDBY をクリアします。これはデフォルトです。

`subnet`
インタフェースのサブネット *address* を設定します。

`tdst` *tunnel_dest_address*
トンネルの着信先アドレスを設定します。このアドレスをトンネルの dest_address と同じ値にするべきではありません。そのようなトンネル経由でシステムを離れるパケットは存在しないからです。

このオプションは廃止され、dladm(1M) の create-iptun および modify-iptun サブコマンドで置き換えられました。

`thoplimit` *n*
トンネルインタフェースのホップ制限を設定します。IPv4 内 IPv6 および IPv4 内 IPv4 トンネルの場合、このホップ制限値は IPv4 ヘッダー内の TTL として使用されます。IPv6 内 IPv6 および IPv6 内 IPv4 トンネルの場合、このホップ制限値は IPv6 ヘッダー内のホップ制限として使用されます。このオプションは単に、ベースとなる IP トンネルリンクの hoplimit リンクプロパティーを変更します (dladm(1M) を参照)。

このオプションは廃止され、dladm(1M) の encaplimit リンクプロパティーで置き換えられました。

`token` *address/prefix_length*
アドレス自動構成で使用される、インタフェースの IPv6 トークンを設定します。

```
example% ifconfig eri0 inet6 token ::1/64
```

`trailers`
このフラグは以前、一部のリンクレベルで IPv4 パケットの非標準カプセル化を引き起こしていました。このリリースに付属するドライバは、このフラグを使用しません。これは互換性のために提供されていますが、無視されます。

`-trailers`
「trailer」リンクレベルカプセル化の使用を無効にします。

`tsrc` *tunnel_src_address*
トンネルの発信元アドレスを設定します。これは、外側のカプセル化している IP ヘッダーの発信元アドレスです。これは、`ifconfig` ですでに構成済みの別のインタフェースのアドレスである必要があります。

このオプションは廃止され、`dladm(1M)` の `create-iptun` および `modify-iptun` サブコマンドで置き換えられました。

`unplumb`
物理インタフェースまたは IPMP インタフェースの場合、関連する論理 IP インタフェースをすべて削除し、IP がそのインタフェースを使用するために必要な plumb をすべて破棄します。IPMP IP インタフェースの場合、グループが空でなければこのコマンドは失敗します。論理インタフェースの場合、その論理インタフェースが削除されます。

インタフェースは、*address_family* パラメータに従って IPv4 用および IPv6 用 (未指定の場合は IPv4 用) に個別に unplumb する必要があります。処理が成功すると、そのインタフェース名が `ifconfig -a` の出力に表示されなくなります。

`up`
論理インタフェースを `UP` としてマークします。その結果、IP モジュールは、関連するアドレス宛てのパケットを受け入れ (そのアドレスがゼロでないかぎり)、さらに関連するすべてのマルチキャストおよびブロードキャスト IP アドレス宛てのパケットも受け入れます。同様に、IP モジュールは、関連するアドレスを送信元アドレスに持つパケットを送信できるようにします。論理インタフェースが少なくとも 1 つ `UP` 状態になっていないと、関連する物理インタフェースでパケットの送受信を行うことができません。

`usesrc` [*name* | `none`]
特定の物理インタフェースを発信元アドレス選択用として指定します。キーワード `none` を使用した場合、それまでの選択がすべてクリアされます。

アプリケーションが `bind(3SOCKET)` を使って 0 以外の発信元アドレスを選択しなかった場合、送信インタフェースとアドレス選択規則 (`ipaddrsel(1M)` を参照) に基づいて、システムが適切な発信元アドレスを選択します。

`usesrc` が指定され、そこで指定されたインタフェースが転送テーブル内で出力用として選択されていた場合、システムは発信元アドレスの選択時に、まず、その指定された物理インタフェースとそれに関連付けられた論理インタフェースを調べます。転送テーブル内に使用可能なアドレスが 1 つも見つからなかった場合には、通常の選択規則が適用されます。たとえば、次のように入力したとします。

```
# ifconfig eri0 usesrc vni0
```

...ここで、`vni0` にアドレス 10.0.0.1 が割り当てられているとすると、システムは、`eri0` 経由で送信されるローカル接続からのすべてのパケットに対

し、10.0.0.1 を発信元アドレスとして優先的に使用します。その他の例については、「使用例」セクションを参照してください。

どのような物理インタフェースでも (ループバックでさえも) 指定できますが、仮想 IP インタフェース (vni(7d) を参照) を指定することもできます。仮想 IP インタフェースは、どの物理ハードウェアにも関連付けられていないため、ハードウェア障害の影響を一切受けません。1 つの仮想インタフェース上にホストされた発信元アドレスを、任意の数の物理インタフェースで使用するよう指定できます。これにより、経路指定に基づくマルチパス化の構成が単純化されます。物理インタフェースの 1 つで障害が発生した場合、正常に機能している残りの物理インタフェースのいずれかを介して通信が継続されます。このシナリオは、仮想インタフェース上にホストされたアドレスの到達可能性が、経路指定プロトコルを使用するなど、何らかの方法を使って通知されていることを前提にしています。

`ifconfig` の `preferred` オプションは、すべてのインタフェースに適用されるため、`usesrc` オプションよりも粗粒度であると言えます。それは、`usesrc` や `setsrc` (route サブコマンド) によって、その順番で上書きされます。

`xmit`
特定の論理インタフェースのパケット送信機能を有効にします。これは、論理インタフェースが「up」状態にある場合のデフォルト動作です。

`-xmit`
特定のインタフェース上でパケット送信を無効にします。そのインタフェースは、パケットの受信は引き続き行います。

`zone` *zonename*
論理インタフェースをゾーン *zonename* 内に配置します。指定されたゾーンは、カーネル内でアクティブになっており、準備完了か実行中の状態になっていなければいけません。ゾーンが停止またはリブートすると、インタフェースは unplumb されます。ゾーンは共有 IP ゾーンとして構成されている必要があります。排他的 IP ゾーンにネットワークインタフェース名を割り当てるには、`zonecfg(1M)` を使用します。

`-zone`
IP インタフェースを大域ゾーン内に配置します。これはデフォルトです。

オペランド

ここでは、*interface* オペランドとそれに影響を与えるアドレスパラメータについて説明します。

interface
文字列。次のいずれかの形式になります。

- *name physical-unit*。例: `eri0`、`ce1`
- *name physical-unit*:*logical-unit*。例: `eri0:1`
- 暗黙的な IP トンネルリンクの場合は、`ip.tun`*N*、`ip6.tun`*N*、または `ip6to4.tun`*N*

ダッシュ (-) で始まるインタフェース名は、インタフェース群を表すオプションがいくつか組み合わさったものとして解釈されます。そのような場合、-a は必ずオプションに含まれている必要がありますが、それ以外の次の追加オプションは、どれでも任意の順序で追加できます。これらのインタフェース名のいずれかが指定された場合、それ以降のコマンドは、条件に一致するすべてのインタフェースに対して適用されます。

-a
指定されたアドレスファミリのすべてのインタフェースに対して、コマンドを適用します。コマンド行からも /etc/default/inet_type 経由でもアドレスファミリが指定されなかった場合、すべてのアドレスファミリが選択されます。

-d
システム内のすべての「down」インタフェースに対して、コマンドを適用します。

-D
DHCP (動的ホスト構成プロトコル) の制御下にないすべてのインタフェースに対して、コマンドを適用します。

-u
システム内のすべての「up」インタフェースに対して、コマンドを適用します。

-Z
ユーザーのゾーン内のすべてのインタフェースに対して、コマンドを適用します。

-4
すべての IPv4 インタフェースに対して、コマンドを適用します。

-6
すべての IPv6 インタフェースに対して、コマンドを適用します。

address_family
アドレスファミリを指定するには、*address_family* パラメータを使用します。ifconfig コマンドが現時点でサポートしているファミリは、inet と inet6 です。アドレスファミリが指定されなかった場合のデフォルトは、inet です。

ifconfig は、インタフェースの情報を表示する際に、/etc/default/inet_type ファイル内の DEFAULT_IP 設定に従います。DEFAULT_IP が IP_VERSION4 に設定されていた場合、ifconfig は、IPv6 インタフェースに関する情報を省略します。ただし、ユーザーがアドレスファミリ (inet、inet6 のいずれか) を ifconfig コマンド行で明示的に指定した場合、そのコマンド行のほうが DEFAULT_IP 設定よりも優先されます。

address
IPv4 ファミリ (inet) の場合、*address* は、ホスト名データベース (hosts(4) を参照) 内またはネットワーク情報サービス (NIS) のマップ hosts 内に存在しているホスト名、インターネット標準の「ドット表記」で表現された IPv4 アドレス、のいずれかになります。

IPv6 ファミリ (inet6) の場合、*address* は、ホスト名データベース (hosts(4) を参照) 内またはネットワーク情報サービス (NIS) のマップ ipnode 内に存在するホスト名、インターネット標準のコロン区切り 16 進形式で表現された IPv6 アドレス、のいずれかになります。後者は*x:x:x:x:x:x:x:x* として表現されます。ここで、*x* は 0 から FFFF までの 16 進数です。

prefix_length
IPv4 ファミリと IPv6 ファミリ (inet と inet6) の場合、*prefix_length* は、0 からアドレス内のビット数までの数値です。アドレス内のビット数は、inet の場合は 32、inet6 の場合は 128 です。*prefix_length* は、ネットマスク内の先頭のセットビットの数を表します。

dest_address
address パラメータのほかに *dest_address* パラメータが指定された場合、そのアドレスは、ポイントツーポイントリンクの他端に位置する対応するインタフェースのアドレスを表します。

tunnel_dest_address
構成中のトンネル以外の特定のインタフェースから到達可能、または到達可能と予想されるアドレス。これによって、トンネルパケットの送信先をトンネルに指示します。このアドレスは、構成中のインタフェース着信先アドレスと同じであってはいけません。

tunnel_src_address
ifconfig を使って「up」として構成された構成済みインタフェースに割り当てられるアドレス。

インタフェースフラグ

ifconfig コマンドがサポートするインタフェースフラグは、次のとおりです。この文脈では、「アドレス」という用語は eri0:0 などの論理インタフェースを表します。一方、「インタフェース」は eri0 などの物理インタフェースを表します。

ADDRCONF
このアドレスは、ステートレス addrconf からのものです。ステートレスメカニズムを使えば、ホストは、ルーターが通知する情報とローカルで利用可能な情報とを組み合わせて固有のアドレスを生成できます。ルーターは、そのリンクに関連付けられたサブネットを識別するプレフィックスを通知します。一方、ホストは、サブネット内のインタフェースを一意に識別する「インタフェース識別子」を生成します。ルーターからの情報が存在しない場合、ホストはリンクローカルアドレスを生成できます。このフラグは IPv6 に固有です。

ANYCAST
: anycast アドレスを示します。anycast アドレスは、あるタイプのサービスを提供する特定のシステムグループの最近傍メンバーを識別します。anycast アドレスは、特定のシステムグループに割り当てられます。パケットは、anycast アドレスで識別される最近傍グループメンバーに配信されます。グループのすべてのメンバーに配信されるのではありません。

BROADCAST
: この broadcast アドレスは有効です。このフラグと POINTTOPOINT は互いに排他です。

CoS
: このインタフェースは、何らかの形式の CoS (Class of Service) マーキングをサポートしています。一例として、VLAN インタフェース上でサポートされる 802.1D ユーザー優先順位マーキングなどが挙げられます。IPMP IP インタフェースの場合、これが設定されるのは、グループ内のすべてのインタフェースで CoS が設定されている場合だけです。

  このフラグは、VLAN リンク経由、および dladm(1M) tagmode リンクプロパティーが normal に設定された Ethernet リンク経由のインタフェース上でのみ設定されることに注意してください。

DEPRECATED
: このアドレスは非推奨です。このアドレスが送信パケットの発信元アドレスとして使用されるのは、このインタフェース上で利用可能なアドレスがほかに存在しない場合と、アプリケーションがこのアドレスに明示的にバインドされている場合だけです。IPv6 の非推奨アドレスは、IPv6 での再採番を行うための標準メカニズムの一部であり、最終的には使用中でないときに削除されます。IPv4 と IPv6 のどちらの場合も、すべての NOFAILOVER アドレスにも DEPRECATED が設定されます。ただし、この仕様は将来のリリースで変更される可能性があります。

DHCPRUNNING
: 論理インタフェースのアドレスは dhcpagent(1M) によって管理されます。IPv6 でそのアドレスがゼロ番目の論理インタフェースにも設定されるのは、そのインタフェースで DHCPv6 が開始された場合です。in.ndpd(1M) を参照してください。

DUPLICATE
: 論理インタフェースに構成されている IP アドレスが重複アドレスなので、このインタフェースは無効にされています。ネットワーク上のほかのノードがこのアドレスを使用しています。これが DHCP によって構成されたアドレスまたは一時アドレスの場合は、可能であれば別のアドレスが自動的に選択されます。それ以外の場合、システムはこのアドレスの回復を定期的に試み、ネットワーク上の衝突が解消されるとインタフェースは回復します。アドレスやネットマスクを変更したり、論理インタフェースを「up」に設定すると、重複検出が再度開始されます。インタフェースを「down」に設定すると、回復処理は終了し、DUPLICATE フラグは削除されます。

FAILED
: in.mpathd デーモンは、このインタフェースで障害が発生したものと判定しました。FAILED インタフェースは、IP データトラフィックの送受信には使用されません。これが IPMP グループの物理 IP インタフェースに設定されている場合、IP データトラフィックは、その IPMP グループで使用可能なほかの IP インタフェースを介して引き続き送受信されます。これが IPMP IP インタフェースに設定されている場合は、グループ全体が障害状態であり、そのグループのどのインタフェースでもデータトラフィックを送受信できません。

FIXEDMTU
: -mtu オプションで MTU が設定されました。このフラグは読み取り専用です。このフラグが設定されたインタフェースは固定の MTU 値を持ちますが、その値は、ドライバがリンク MTU 変更を IP に通知する際に発生する可能性のある動的 MTU 変更の影響を受けません。

INACTIVE
: 物理インタフェースは正常に機能していますが、管理ポリシーに従ってデータトラフィックの送受信には使用されていません。このフラグは、最初 standby サブコマンドによって設定され、その後 in.mpathd によって管理されます。またこのフラグは、FAILBACK=no モードが有効化された (in.mpathd(1M) を参照) 場合に、修復された IP インタフェースがまだ使用されていないことを示すためにも設定されます。

IPMP
: これが IPMP IP インタフェースであることを示します。

LOOPBACK
: これがループバックインタフェースであることを示します。

MULTI_BCAST
: このインタフェースではブロードキャストアドレスがマルチキャスト用として使用されることを示します。

MULTICAST
: このインタフェースはマルチキャストをサポートします。IP は、ハードウェアブロードキャストをサポートするインタフェース、つまりポイントツーポイントリンクインタフェースは、マルチキャストをサポートするものと仮定します。

NOARP
: ブロードキャストアドレスを持たないデバイスのすべてのインタフェースに対応した、このインタフェース用のアドレス解決プロトコル (ARP) が存在しません。このフラグは IPv4 に固有です。

NOFAILOVER
: この論理インタフェースに関連付けられたアドレスは、関連する物理 IP インタフェースのプローブベース障害検出を行う際に in.mpathd から使用できます。

NOLOCAL
このインタフェースはアドレスを持ちません。持つのはオンリンクサブネットだけです。

NONUD
このインタフェースでは、NUD が無効になっています。各ノードは NUD (近傍不到達検出) を使って、自身が能動的にパケットを送信する近傍ノードの到達状態を追跡し、到達不可能な近傍ノードが検出された際に回復処理を実行します。このフラグは IPv6 に固有です。

NORTEXCH
このインタフェースは経路指定情報を交換しません。RIP-2 の場合、経路指定パケットがこのインタフェース経由で送信されることはありません。さらに、このインタフェースを経由したと思われるメッセージには、応答が返されません。このインタフェースのサブネットまたはアドレスが、ほかのインタフェース経由でのほかのルーター宛の通知に含められることはありません。

NOXMIT
このアドレスがパケットを送信しないことを示します。また、RIP-2 はこのアドレスを通知しません。

OFFLINE
インタフェースはオフライン状態なので、IP データトラフィックを送受信できません。これは、IPMP グループ内の IP インタフェースにしか設定されません。if_mpadm(1M) および cfgadm(1M) を参照してください。

POINTOPOINT
このアドレスがポイントツーポイントリンクであることを示します。このフラグと BROADCAST は互いに排他です。

PREFERRED
このアドレスは優先 IPv6 発信元アドレスです。このアドレスは、すべての IPv6 着信先との IPv6 通信の発信元アドレスとして使用されます。ただし、システム上の別のアドレスがより適切なスコープを持つ場合は除外します。DEPRECATED フラグは PREFERRED フラグよりも優先されます。

PRIVATE
このアドレスが通知されないことを示します。RIP-2 の場合、このインタフェースは通知を送信する際に使用されます。ただし、サブネットやこのアドレスが、ほかのルーター宛の通知に含められることはありません。

PROMISC
インタフェースがプロミスキュアスモードになっていることを示す、読み取り専用のフラグ。プロミスキュアスモードのインタフェースに関連付けられたアドレスでは必ず、PROMISC フラグが表示されます (ifconfig -a への応答時など)。

ROUTER
このインタフェースへの、およびこのインタフェースからの、IP パケットの転送を行えることを示します。

RUNNING
インタフェースが必要とするリソースが確保されていることを示します。一部のインタフェースでは、これは、リンクが稼働していることも示します。IPMP IP インタフェースの場合、グループ内の 1 つの IP インタフェースがアクティブになっているかぎり、RUNNING が設定されます。

STANDBY
IPMP グループ内の別のインタフェースが使用不可能にならないかぎり、この物理インタフェースがデータトラフィックに使用されないことを示します。INACTIVE および FAILED フラグは、インタフェースがアクティブに使用されているかどうかを示します。

TEMPORARY
これがRFC 3041 に規定された一時 IPv6 アドレスであることを示します。

UNNUMBERED
このフラグが設定されるのは、このリンクのローカル IP アドレスが、システム内のほかのリンクのローカルアドレスに一致した場合です。

UP
論理インタフェース (と関連する物理インタフェース) が動作していることを示します。IP モジュールは、UP アドレス宛てのパケットを受け入れ (そのアドレスがゼロでないかぎり)、さらに関連するすべてのマルチキャストおよびブロードキャスト IP アドレス宛てのパケットも受け入れます。同様に、IP モジュールは、UP アドレスを送信元アドレスに持つパケットを送信できるようにします。

VIRTUAL
この物理インタフェースには基本となるハードウェアがないことを示します。仮想インタフェースを通してパケットの送信や受信を行うことはできません。このようなインタフェースは、複数のインタフェース上で使用できるローカルアドレスを構成する場合に役立ちます。(usesrc オプションも参照してください。)

L3PROTECT
dladm(1M) の allowed-ips リンクプロパティーを使用して、物理インタフェース上でレイヤー 3 保護が適用されたことを示します。

PROBER
IPMP グループ内のベースとなる FAILED インタフェースの修復が完了したかどうかを判定するための検査を行なっていることを示します。PROBER フラグとそのセマンティクスは、Solaris IPMP 実装の中身に依存するため、変更される可能性があります。

論理インタフェース

Solaris の TCP/IP では、1 つの物理ネットワークインタフェースに複数の論理インタフェースを関連付けることができます。これにより、ネットワークインタフェースを 1 つしか持たないようなマシンにも、複数の IP アドレスを割り当てることが可能になります。物理ネットワークインタフェースが *driver-name physical-unit-number* という形式の名前を持つのに対し、論理インタフェースは *driver-name*

physical-unit-number:*logical-unit-number* という形式の名前を持ちます。特定の物理インタフェースを構成してシステムに組み込むには、`plumb` コマンドを使用します。例:

```
example% ifconfig eri0 plumb
```

物理インタフェースの「plumb」が完了すると、その物理インタフェースに関連付けられた論理インタフェースを構成できるようになります。それには、`-plumb` または `-addif` オプションを指定して再度 `ifconfig` コマンドを実行します。

```
example% ifconfig eri0:1 plumb
```

これは、物理インタフェース `eri0` に関連付けられた特定の論理インタフェースを割り当てます。次のコマンドを見てください。

```
example% ifconfig eri0 addif 192.168.200.1/24 up
```

これは、`eri0` 物理インタフェース上で利用可能な次の論理ユニット番号を割り当てるとともに、*address* と *prefix_length* も割り当てます。

論理インタフェースは、関連付けられた物理インタフェースとは異なるパラメータ (*address*、*prefix_length* など) を使って構成することができます。また、同じ物理インタフェースに関連付けられた複数の論理インタフェースにそれぞれ異なるパラメータを与えることもできます。各論理インタフェースは、既存の「up」状態の物理インタフェースに関連付ける必要があります。したがって、たとえば、論理インタフェース `eri0:1` を構成できるのは、物理インタフェース `eri0` を plumb し終わったあとです。

論理インタフェースを削除するには、`unplumb` または `removeif` オプションを使用します。たとえば、次のように指定します。

```
example% ifconfig eri0:1 down unplumb
```

これは、論理インタフェース `eri0:1` を削除します。

Ip マルチパスグループ

同じリンクレイヤーブロードキャストドメインを共有する物理インタフェースは、`group` サブコマンドを使用して単一の IP マルチパス (IPMP) グループ内に集める必要があります。各 IPMP グループには IPMP IP インタフェースが関連付けられます。このインタフェースは、`ipmp` サブコマンドを使用して明示的に作成する (推奨される方法) ことも、`ifconfig` で IP インタフェースを新しい IPMP グループ内に配置することによって暗黙的に作成することもできます。暗黙的に作成される IPMP インタフェースには `ipmp`*N* という名前が付けられます。ここで *N* は、既存の IP インタフェース名や IPMP グループ名と競合しない最小の整数です。

各 IPMP IP インタフェースの作成時には対応する IPMP グループ名が使用されますが、このグループ名は、`group` サブコマンドを使用すると変更できます。各 IPMP IP インタフェースは、高可用性の一連の IP アドレスをホストします。それらのアドレスは、グループ内の少なくとも 1 つのインタフェースがアクティブであるかぎり、到達可能な状態に保たれます。なお、この場合の「アクティブ」は、少なくと

も 1 つの UP アドレスが存在し、かつ INACTIVE、FAILED、および OFFLINE がクリアされた状態として定義されます。IPMP IP インタフェース上でホストされる IP アドレスは、静的に構成することも、dhcp サブコマンドを使用して DHCP 経由で構成することもできます。

同じ IPMP グループに割り当てられたインタフェースはすべて同等とみなされ、in.mpathd によって障害発生の有無が監視されます。グループ内にアクティブなインタフェースが残っている場合、IP インタフェースの障害 (と後続のすべての修復) が、ソケットベースのアプリケーションに対して透過的に処理されます。さらに、IPMP は動的再構成フレームワーク (cfgadm(1M)を参照) と統合されています。このフレームワークを使用すると、ソケットベースのアプリケーションに対して透過的な方法で、ネットワークアダプタを交換することができます。

IP モジュールは自動的に、IPMP グループ内のすべてのアクティブインタフェースにすべてのアウトバウンドトラフィックを負荷分散します。同様に、IPMP IP インタフェースでホストされているすべての UP アドレスが複数のアクティブインタフェースに分散され、インバウンド負荷分散が促進されます。ipmpstat(1M) ユーティリティーを使用すると、IP データアドレスと IP インタフェースの現在のバインドなど、IPMP サブシステムのさまざまな側面を監視することができます。

あるインタフェースが IPMP グループ内に配置されると、次の場合を除き、UP 状態の論理インタフェースがすべて、そのグループで使用できるように IPMP IP インタフェースに「移行」されます。

- 論理インタフェースが NOFAILOVER とマークされている。
- 論理インタフェースが IPv6 リンクローカルアドレスをホストしている。
- 論理インタフェースが IPv4 0.0.0.0 アドレスをホストしている。

同様に、インタフェースがいったんグループ内に配置されると、ある論理インタフェースが、UP 状態でかつ上記のいずれかの条件によって除外されないように変更された場合、その論理インタフェースも関連する IPMP IP インタフェースに移行されます。アドレスを提供した物理インタフェースがグループから削除された場合でも、その論理インタフェースの移行が元に戻されることはありません。

IPMP グループ内に配置する各インタフェースは、in.mpathd がプローブベースの障害検出で使用する「テスト」アドレスで構成することもできます。in.mpathd(1M) を参照してください。これらのアドレスは、UP とマークされる前に (-failover サブコマンドを使用して) NOFAILOVER とマークされる必要があります。テストアドレスは、dhcp サブコマンドを使用して DHCP 経由で取得することもできます。

IPMP のより詳しい背景情報については、『Oracle Solaris 管理: ネットワークインタフェースとネットワーク仮想化』 の IPMP に関する章を参照してください。

Ipv6 インタフェースの構成

ifconfig による IPv6 物理インタフェースの plumb と「up」構成が完了すると、そのインタフェースには、IPv6 リンクローカルアドレスが自動的に割り当てられます。このアドレスの最後の 64 ビットは、そのインタフェースの MAC アドレスから計算されます。

```
example% ifconfig eri0 inet6 plumb up
```

次の例は、リンクローカルアドレスのプレフィックスが fe80::/10 であることを示しています。

```
example% ifconfig eri0 inet6
ce0: flags=2000841<UP,RUNNING,MULTICAST,IPv6>
          mtu 1500 index 2
        inet6 fe80::a00:20ff:fe8e:f3ad/10
```

リンクローカルアドレスは、ローカルサブネット上での通信にのみ使用され、ほかのサブネットからは見えません。

プレフィックスを通知しているリンク上に通知 IPv6 ルーターが存在する場合、新しく plumb された IPv6 インタフェースは、1 つ以上の論理インタフェースをそのプレフィックス通知に基づいて自動構成します。たとえば、プレフィックス通知が 2001:0db8:3c4d:0:55::/64 であった場合、自動構成されたインタフェースは次のようになります。

```
eri0:2: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6>
          mtu 1500 index 2
        inet6 2001:0db8:3c4d:55:a00:20ff:fe8e:f3ad/64
```

プレフィックス通知がリンク上に存在しない場合でも、グローバルアドレスを手動で割り当てることができます。例:

```
example% ifconfig eri0 inet6 addif \
2001:0db8:3c4d:55:a00:20ff:fe8e:f3ad/64 up
```

IP 上の IP トンネルインタフェースの構成

IP トンネルは、概念的に 2 つの部分で構成されます。1 つは、2 つ以上の IP ノード間の仮想リンク、もう 1 つはこのリンク上の IP インタフェースです。このインタフェースを使用すると、基礎となるリンクによってカプセル化された IP パケットをシステムで送受信できます。

トンネルリンクの構成には dladm(1M) コマンドを使用し、それらのトンネルリンク上での IP インタフェースの構成には ifconfig を使用します。IPv4 上の IPv4 トンネルを作成するには、IPv4 トンネルリンク上で IPv4 インタフェースを plumb します。IPv4 上の IPv6 トンネルを作成するには、IPv6 トンネルリンク上で IPv4 インタフェースを plumb します。

IP トンネルリンク上で IPv6 インタフェースを plumb すると、その IPv6 アドレスが自動的に設定されます。IPv4 および IPv6 トンネルでは、fe80::*interface-id* という形式の送信元および送信先リンクローカルアドレスが構成されます。IPv4 トンネルの場合、*interface-id* は IPv4 トンネルの送信元または送信先アドレスになります。IPv6 の場合、*interface-id* は、IPv6 トンネルの送信元または送信先アドレスの下位 64 ビットになります。たとえば、10.1.2.3 と 10.4.5.6 との間の IPv4 トンネルの場合、IPv6 インタフェースの IPv6 リンクローカルの送信元アドレスと送信先アドレスは、fe80::a01:203 と fe80::a04:506 になります。2000::1234:abcd と 3000::5678:abcd との間の IPv6 トンネルの場合、インタフェースの IPv6 リンク

ローカルの送信元アドレスと送信先アドレスは、fe80::1234:abcd と fe80::5678:abcd になります。これらのデフォルトのリンクローカルアドレスを上書きするには、ほかのポイントツーポイントインタフェースと場合と同じく、アドレスを明示的に指定します。

6to4 トンネルの場合、2002:*tsrc*::1/16 という形式の 6to4 グローバルアドレスが構成されます。*tsrc* の部分はトンネルの送信元 IPv4 アドレスです。6to4 インタフェースの接頭辞長は自動的に 16 に設定されますが、これは、すべての 6to4 パケット (送信先が 2002::/16 の範囲のもの) が 6to4 トンネルインタフェースに転送されるからです。たとえば、トンネル送信元が 75.1.2.3 の 6to4 リンクでは、IPv6 インタフェースのアドレスは 2002:4b01:203::1/16 になります。

追加の IPv6 アドレスを追加するには、addif オプションを使用するか、追加の論理インタフェースを plumb します。

下位互換性を保つため、特殊な名前を使用してトンネル IP インタフェースを plumb すると、dladm create-iptun を呼び出さなくても、トンネルリンクが暗黙的に作成されます。それらのトンネル名を次に示します。

ip.tun*N*	IPv4 トンネル
ip6.tun*N*	IPv6 トンネル
ip.6to4tun*N*	6to4 トンネル

これらのトンネルは「暗黙的なトンネル」であり、dladm show-iptun 出力では i フラグで示されます。これらの特殊 IP インタフェースの plumb 先となるトンネルリンクは自動的に作成され、最後の参照が解放された (つまり最後の IP インタフェースが unplumb された) 時点で自動的に削除されます。

ifconfig の tsrc、tdst、encaplim、および hoplimit オプションは廃止済みであり、下位互換性を保つためだけに維持されています。これらは、dladm(1M) の対応するオプションと同等です。

トンネルのセキュリティー設定の表示

IP トンネルインタフェースの ifconfig 出力には、ベースとなる IP トンネルリンクで IPsec ポリシーが構成されているかどうかが表示されます。たとえば、IPsec ポリシーが存在する場合には、次のような形式の行が表示されます。

```
tunnel security settings  -->  use 'ipsecconf -ln -i ip.tun1'
```

ifconfig または ipsecconf(1M) を使用してセキュリティーポリシーの設定を行なっていない場合、トンネルのセキュリティー設定は表示されません。

使用例

例 1　ifconfig コマンドの使用

ワークステーションが Ethernet に接続されていない場合、そのネットワークインタフェース (たとえば eri0 など) を次のようにして「down」としてマークしてください。

例 1　ifconfig コマンドの使用　　*(続き)*

```
example% ifconfig eri0 down
```

例 2　アドレス指定情報の出力

各インタフェースのアドレス指定情報を出力するには、次のコマンドを使用します。

```
example% ifconfig -a
```

例 3　ブロードキャストアドレスのリセット

ネットマスクが正しく設定された状態で、各インタフェースのブロードキャストアドレスをリセットするには、次のコマンドを使用します。

```
example% ifconfig -a broadcast +
```

例 4　Ethernet アドレスの変更

インタフェース ce0 の Ethernet アドレスを変更するには、次のコマンドを使用します。

```
example% ifconfig ce0 ether aa:1:2:3:4:5
```

例 5　IP 内 IP トンネルの構成

IP 内 IP トンネルを構成するには、まず、次のように IP トンネルリンクを作成します (tunsrc と tundst は、/etc/hosts 内に対応する IPv4 エントリを持つホスト名)。

```
example% dladm create-iptun -T ipv4 -s tunsrc -d tundst tun0
```

続いて次のように、送信元および送信先アドレスを指定してポイントツーポイントインタフェースを plumb します (mysrc と thedst は、/etc/hosts 内に対応する IPv4 エントリを持つホスト名)。

```
example% ifconfig tun0 plumb mysrc thedst up
```

トンネルのセキュリティープロパティーを設定するには、前述の ipsecconf(1M) を使用します。

IPv6 トンネルを構成するには、create-iptun でトンネルタイプ ipv6 を使用します。どちらかのタイプのトンネル上で、IPv6 インタフェースを plumb することもできます。

例 6　6to4 トンネルの構成

6to4 トンネルを構成するには、まず、次のように 6to4 トンネルリンクを作成します (myv4addr は、/etc/hosts 内に対応する IPv4 エントリを持つホスト名)。

```
example% dladm create-iptun -T 6to4 -s myv4addr my6to4tun0
```

例 6　6to4 トンネルの構成　　*(続き)*

続いて、次のように、このリンク上で IPv6 インタフェースを plumb します。

```
example% ifconfig my6to4tun0 inet6 plumb up
```

このインタフェースの IPv6 アドレスは、前述したように自動的に設定されます。

例 7　インタフェース上の IP 転送の構成

単一インタフェース上の IP 転送を有効にするには、次のコマンドを使用します。

```
example% ifconfig eri0 router
```

単一インタフェース上の IP 転送を無効にするには、次のコマンドを使用します。

```
example% ifconfig eri0 -router
```

例 8　仮想インタフェースを使用した発信元アドレス選択の構成

次のコマンドは発信元アドレス選択を構成することで、発信元アドレスが関連付けられずにローカルで生成されて qfe2 から出力されるすべてのパケットに対し、vni0 上にホストされた発信元アドレスが優先的に使用されるようにします。

```
example% ifconfig qfe2 usesrc vni0
```

qfe2 および vni0 インタフェースに対する ifconfig -a の出力は、次のようになります。

```
qfe2: flags=1100843<UP,BROADCAST,RUNNING,MULTICAST,ROUTER,IPv4> mtu
  1500 index 4
  usesrc vni0
  inet 1.2.3.4 netmask ffffff00 broadcast 1.2.3.255
  ether 0:3:ba:17:4b:e1
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
  mtu 0 index 5
  srcof qfe2
  inet 3.4.5.6 netmask ffffffff
```

前述の ifconfig 出力に含まれる usesrc と srcof キーワードに注目してください。これらのキーワードは、物理インタフェース単位のパラメータであるにもかかわらず、物理インタフェースの論理インスタンス上にも現れます。構成元のインタフェースに対する ifconfig には、srcof キーワードは現れません。この情報は、usesrc が設定された一連のインタフェースから自動的に決定されます。

none キーワードを使用した次のコマンドは、前述の ifconfig usesrc コマンドによる効果を取り消します。

```
example% ifconfig qfe2 usesrc none
```

例 8 仮想インタフェースを使用した発信元アドレス選択の構成 (続き)

このコマンドの実行後、ifconfig -a の出力は次のようになります。

```
qfe2: flags=1100843<UP,BROADCAST,RUNNING,MULTICAST,ROUTER,IPv4> mtu
  1500 index 4
  inet 1.2.3.4 netmask ffffff00 broadcast 1.2.3.255
  ether 0:3:ba:17:4b:e1
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
  mtu 0 index 5
  inet 3.4.5.6 netmask ffffffff
```

前述の出力には usesrc と srcof キーワードは含まれていません。

例 9 IPv6 アドレスの発信元アドレス選択の構成

次のコマンドは、vni0 上にホストされた発信元アドレスを選択することで、IPv6 アドレスの発信元アドレス選択を構成します。

```
example% ifconfig qfe1 inet6 usesrc vni0
```

このコマンドの実行後、ifconfig -a の出力は次のようになります。

```
qfe1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 3
  usesrc vni0
  inet6 fe80::203:baff:fe17:4be0/10
  ether 0:3:ba:17:4b:e0
vni0: flags=2002210041<UP,RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
  index 5
  srcof qfe1
  inet6 fe80::203:baff:fe17:4444/128
vni0:1: flags=2002210040<RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
  index 5
  srcof qfe1
  inet6 fec0::203:baff:fe17:4444/128
vni0:2: flags=2002210040<RUNNING,NOXMIT,NONUD,IPv6,VIRTUAL> mtu 0
  index 5
  srcof qfe1
  inet6 2000::203:baff:fe17:4444/128
```

qfe1 から出力されるパケットの着信先のスコープに応じて、適切なスコープを持つ発信元アドレスが vni0 とその別名から選択されます。

例 10 共有 IP ゾーンでの発信元アドレス選択の使用

次の例は、Solaris の zones(5) 機能における usesrc 機能の使用方法を示したものです。大域ゾーン内で次の各コマンドを呼び出したとします。

例 10　共有 IP ゾーンでの発信元アドレス選択の使用　　(続き)

```
example% ifconfig hme0 usesrc vni0
example% ifconfig eri0 usesrc vni0
example% ifconfig qfe0 usesrc vni0
```

前述のコマンドの実行後、その仮想インタフェースに対する ifconfig -a の出力は次のようになります。

```
vni0: flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL>
   mtu 0 index 23
   srcof hme0 eri0 qfe0
   inet 10.0.0.1 netmask ffffffff
vni0:1:
   flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
   index 23
   zone test1
   srcof hme0 eri0 qfe0
   inet 10.0.0.2 netmask ffffffff
vni0:2:
   flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
   index 23
   zone test2
   srcof hme0 eri0 qfe0
   inet 10.0.0.3 netmask ffffffff
vni0:3:
   flags=20011100c1<UP,RUNNING,NOARP,NOXMIT,ROUTER,IPv4,VIRTUAL> mtu 0
   index 23
   zone test3
   srcof hme0 eri0 qfe0
   inet 10.0.0.4 netmask ffffffff
```

仮想インタフェースの別名が、ゾーン (test1、test2、および test3) ごとに 1 つずつ存在しています。同じゾーン内の仮想インタフェース別名に含まれる発信元アドレスが選択されます。これらの仮想インタフェース別名は、次のように zonecfg(1M) を使って作成されたものです。

```
example% zonecfg -z test1
zonecfg:test1> add net
zonecfg:test1:net> set physical=vni0
zonecfg:test1:net> set address=10.0.0.2
```

test2 ゾーンと test3 ゾーンのインタフェースとアドレスも同じ方法で作成されます。

例 11　DHCPv6 を無効にする

次の例は、すべてのインタフェースで DHCPv6 の自動使用を無効にし、hme0 というインタフェースの DHCPv6 をただちに停止する方法を示したものです。DHCPv6 の自動構成メカニズムの詳細については、in.ndpd(1M) および ndpd.conf(4) を参照してください。

```
example% echo ifdefault StatefulAddrConf false >> /etc/inet/ndpd.conf
example% pkill -HUP -x in.ndpd
example% ifconfig hme0 dhcp release
```

ファイル

/etc/netmasks
　ネットマスクのデータ。

/etc/default/inet_type
　デフォルトのインターネットプロトコルタイプ。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
コマンド行オプションに対するインタフェースの安定性	確実
コマンド出力に対するインタフェースの安定性	不確実

関連項目

dhcpinfo(1), cfgadm(1M), dhcpagent(1M), dladm(1M), if_mpadm(1M), in.mpathd(1M), in.ndpd(1M), in.routed(1M), ipadm(1M), ipmpstat(1M), ipsecconf(1M), ndd(1M), netstat(1M), zoneadm(1M), zonecfg(1M), ethers(3SOCKET), gethostbyname(3NSL), getnetbyname(3SOCKET), hosts(4), inet_type(4), ndpd.conf(4), netmasks(4), networks(4), nsswitch.conf(4), attributes(5), privileges(5), zones(5), arp(7P), ipsecah(7P), ipsecesp(7P)

『Oracle Solaris の管理: IP サービス』

診断

ifconfig は、次のことを示すメッセージを送信します。

- 指定されたインタフェースが存在しないかどうか
- 要求されたアドレスが未知かどうか
- 非特権ユーザーが特定のインタフェースの構成を変更しようとしているかどうか

注意事項

ホスト名を選択する際に、broadcast、down、private、trailers、up やその他のオプションの名前を選択しないでください。これらの名前のいずれかをホスト名として選択した場合、診断が非常に難しい特異な問題が発生する可能性があります。

名前

if_mpadm – administer interfaces in an IP multipathing group

形式

`if_mpadm -d | -r` *ifname*

機能説明

The `if_mpadm` utility administers IP interfaces that are part of an IP Multipathing (IPMP) group. Currently, administration is limited to offlining IP interfaces and undoing previous offline operations.

When an IP interface is taken offline, all IP data traffic that was flowing over the IP interface is moved to another IP interface in the IPMP group. In addition, all `UP` IP addresses hosted on the IP interface are brought down, causing `in.mpathd(1M)` to stop probe-based failure detection on the IP interface. As a result, an offline IP interface will not be used for any inbound or outbound IP traffic. Only IP interfaces that are in an IPMP group may be brought offline. If the IP interface is the last functioning interface in the IPMP group, the offline operation will fail.

When an offline operation is undone, any IP addresses hosted on that IP interface are brought `UP` and will be considered by `in.mpathd` for probe-based failure detection. In addition, provided the IP interface is otherwise active (see `in.mpathd(1M)`), it will again be used to send and receive IP data traffic for the IPMP group. Note that not all offline operations can be undone. For instance, `in.mpathd` may have offlined an IP interface because its hardware address was not unique within its IPMP group. The `ipmpstat` utility can be used to determine why an IP interface is offline, identify which IP interfaces in a group are being used for inbound and outbound IP traffic, and more; see `ipmpstat(1M)`.

オプション

The `if_mpadm` utility supports the following options:

`-d` *ifname*
: Offline the IP interface specified by *ifname*. If *ifname* is not in an IPMP group, or the offline would cause the IPMP group to lose network connectivity, the operation will fail.

`-r` *ifname*
: Undo a previous offline of the IP interface specified by *ifname*. If *ifname* is not offline, the operation will fail.

使用例

例 1 Offlining an IP Interface

The following command offlines the IP interface `under0`, causing any IP packets that were being sent and received through it to be handled by another IP interface in its group.

```
example% if_mpadm -d under0
```

例 2 Undoing a Previous Offline Operation

Use the following command to undo the operation in the previous example:

```
example% if_mpdadm -r under0
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

関連項目

ifconfig(1M), in.mpathd(1M), ipmpstat(1M), attributes(5)

診断

`cannot offline: no other functioning interfaces are in its IPMP group.`

説明: This message means that offlining the IP interface would leave the IPMP group without network connectivity.

`cannot offline: not a physical interface or not in an IPMP group`

説明: This means that the IP interface is not an underlying interface in an IPMP group, and therefore is not eligible to be offlined.

名前

ifparse – parse `ifconfig` command line

形式

`/usr/sbin/ifparse [-fs]` *addr_family commands*

機能説明

Use the `ifparse` command to parse the `ifconfig(1M)` command line options and output substrings, one per line, as appropriate. If no options are specified, `ifparse` returns the entire `ifconfig` command line as a series of substrings, one per line.

オプション

The `ifparse` command supports the following options:

`-f` Lists only substrings of the `ifconfig` command line that are relevant to IP network multipath failover

`-s` Lists only substrings of the `ifconfig` command line that are not relevant to IP network multipath failover

オペランド

The `ifparse` command *does not* support the *interface* operand of the `ifconfig` command.

使用例

例 1 Parsing Command Line Options Relevant to Failover

The following example shows the use of the `ifparse` command to parse the command line options relevant to IP network multipath failover:

```
example# ifparse -f inet 1.2.3.4 up group one addif 1.2.3.5 -failover up
set 1.2.3.4 up
```

例 2 Parsing Command Line Options That Are Not Relevant to Failover

The following example shows the use of the `ifparse` command to parse the command line options that are not relevant to IP network multipath failover:

```
example# ifparse -s inet 1.2.3.4 up group one addif 1.2.3.5 -failover up
group one
addif 1.2.3.5 -failover up
```

例 3 Parsing the Command Line For All Options

The following example shows the use of the `ifparse` command to parse the command line for all `ifconfig` options:

```
example# ifparse inet 1.2.3.4 up group one addif 1.2.3.5 -failover up
group one
set 1.2.3.4 up
addif 1.2.3.5 -failover up
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Obsolete

関連項目

ifconfig(1M), attributes(5)

診断

usage: -fs <addr_family> <commands>
説明: This message indicates an invalid command line.

ifparse: Not enough space
説明: This message indicates insufficient memory.

ifparse: dhcp not supported for inet6
説明: DHCP operations are not supported for the inet6 address family.

ifparse: Operation <operation> not supported for <addr_family>
説明: Most operations cannot be used with all address families. For example, the broadcast operation is not supported on the inet6 address family.

ifparse: no argument for <operation>
説明: Some operations, for example broadcast, require an argument.

注意事項

The ifparse command is classified as an obsolete interface. It will likely be removed in a future release. You should not develop applications that depend upon this interface.

名前

iiadm – Sun StorageTek Availability Suite ポイントインタイムコピー操作を制御するコマンド行インタフェース

形式

```
iiadm -e {ind | shd} master_vol shadow_vol bitmap_vol

iiadm -ne ind master_vol shadow_vol bitmap_vol

iiadm [-p] [-n] {-c | -u} {s | m} volume_set

iiadm [-adDilR] volume_set

iiadm [-p] [-n] -w volume_set

iiadm [-hilLv]

iiadm -P delay units volume_set

iiadm -P volume_set

iiadm -A overflow_vol volume_set

iiadm [-OQ] overflow_vol

iiadm -E volume_set

iiadm [-IJ] volume_set bitmap

iiadm -g group_name [-aAcdDeEilLmPRuw]

iiadm [-C] cluster_tag [options]
```

機能説明

ポイントインタイムコピーソフトウェアは、Solaris オペレーティングシステムのポイントインタイムスナップショット機能です。

ポイントインタイムコピースナップショットは、すぐに使用可能になる、時間を固定し、瞬間的に休止したボリュームの複製ビューです。スナップショットがいったん取得されると、ポイントインタイムコピーソフトウェアは、マスターボリュームとシャドウボリュームの両方のデータへの読み取り/書き込みアクセスを即座に可能にします。

ポイントインタイムコピーソフトウェアは、スナップショット確立以降の (書き込みによって発生した) マスターボリュームとシャドウボリュームとの差異を追跡します。この機能により、シャドウボリュームのデータにアクセスするアプリケーションとは時間的に無関係にマスターボリュームのデータにアクセスするアプリケーションの処理を進めたり、その逆を行なったりすることが可能になります。

マスターボリュームとシャドウボリュームとの差異を追跡するポイントインタイムコピーソフトウェアの機能により、あとで行う高速再同期やフルコピーが容易になります。ボリュームの再同期は、シャドウからマスターへ行うことも、マスターからシャドウへ行うこともできます。

ポイントインタイムが確立または再確立されたすぐあと (CLI プロンプトが復帰するか次のシェルスクリプトコマンドが読み取られたあと) で、マスターボリューム

を再びマウントしたり、マスターボリュームを使用するアプリケーションを再開したりできます。また、シャドウボリュームをマウントしてすぐにアクセスを開始することもできます。

iiadm コマンド行ユーティリティーは、コマンドが呼び出されるたびに 1 つのアクションだけを実行します。このため、複数のオプションを組み合わせることはできません。ただし、次の全体用のコマンド修飾子との組み合わせは例外です。

- 処理項目が 1 つも入力されなかった場合、iiadm は、現在構成されているポイントインタイムコピーセット (中断されていないもの) の一覧を表示します。処理項目が複数入力された場合や入力された処理項目の指定方法が間違っていた場合、iiadm は特定のエラーメッセージと使用法の簡単な要約を標準エラーに表示します。
- ポイントインタイムコピーのオプション ENABLE (-e)、COPY (-c)、および UPDATE (-u) には、関連する 2 つのシャドウボリューム選択修飾子 {ind|dep} があります。これらの修飾子は、作成するポイントインタイムコピーボリュームセットの種類を指定するために使用されます。

独立 (ind) スナップショットの場合、ポイントインタイムコピーソフトウェアがマスターからシャドウへのフルボリュームコピー操作を実行します。コピーが完了すると、シャドウボリュームのデータは、確立時のマスターボリュームのデータと同一になります。独立シャドウは、データの物理コピーが 2 つ必要な場合に作成します。独立シャドウボリュームのサイズは、マスターボリュームのサイズと同じか、それより大きい必要があります。シャドウからマスターへの再同期が検討されている環境では、マスターボリュームとシャドウボリュームのサイズを同じにすることをお勧めします。

依存 (dep) スナップショットの場合、ポイントインタイムコピーソフトウェアはフルボリュームコピーを実行しません。結果として得られるシャドウボリュームは、変更されていないすべてのデータブロックでマスターボリュームに依存します。これらのデータブロックは要求されるまでコピーされません。依存シャドウは、データの物理コピーが 2 つも必要ない場合に作成します。依存シャドウボリュームのサイズは、マスターボリュームと同じか、それより小さくすることができます。マスターより小さいシャドウボリュームは「小型依存シャドウボリューム」と呼ばれ、通常は、ポイントインタイムコピーボリュームセットに対する変更量がマスターボリューム全体のサイズに比べて少ない場合に使用されます。

次の構文を使用すると、エクスポート可能な独立シャドウボリュームを Sun Cluster 環境で作成できます。

```
# iiadm -ne ind master shadow bitmap
```

小型依存シャドウボリュームを使用する場合には問題が 1 つ発生します。ポイントインタイムコピーボリュームセットが有効化される時点で、そのサイズが確定されてしまうという問題です。使用期間中のボリュームセット全体に対する変更量がシャドウボリュームに割り当てられた容量を超えると、シャドウボリュームが容量

不足としてマークされます。容量不足が発生したあとも、その容量不足が発生したデータ部分が要求されないかぎり、シャドウボリュームからデータを読み取ることができます。その要求がいったん発生すると読み取りが失敗し、シャドウボリュームがオフラインとしてマークされます。

ポイントインタイムコピーではこの問題を解決するため、ポイントインタイムコピーの既存の依存ボリュームセットにオーバーフローボリュームを関連付ける機能がサポートされています。したがって、小型依存シャドウボリュームのサイズが小さすぎた場合やボリュームセットに対して予想外の量の変更が発生した場合、変更されたデータを関連オーバーフローボリュームにリダイレクトできます。このオーバーフローボリュームの効率的な使用を促進するため、必要に応じてこのボリュームを複数のポイントインタイムコピーボリュームセットに関連付けられるようになっています。

考慮事項

ポイントインタイムコピーの enable、copy、または update 操作の呼び出し前に、ポイントインタイムコピーは、ファイルシステムのパニックが発生しないように、シャドウボリュームがマウントされていないことを確認します。また、ポイントインタイムスナップショットが取得される短時間の間だけ、マスターボリュームを使用するすべてのアプリケーションをマウント解除または中断 (休止) することをお勧めします。これにより、原子的に一貫性のあるポイントインタイムスナップショットの取得が保証されます。

マスターボリュームがマウント解除ではなく中断された場合には、fsck(1M) などのボリューム検証ユーティリティーを使用して新しいポイントインタイムシャドウボリュームの整合性を検証することをお勧めします。その理由は、ポイントインタイムコピーにより、マウントされたマスターボリュームからマウント解除されたシャドウボリュームへのポイントインタイムコピーが行われているためです。シャドウボリュームのマウント中、ファイルシステムはそのボリュームがマウントされた状態にあることを検出します。通常、この状態が発生するのはシステムがクラッシュした場合だけであるため、ファイルシステムはポイントインタイムコピーが実行されたのではなくシステム障害が発生したものと仮定して、ボリュームの整合性の検証を試みます。

環境オプション

/usr/kernel/drv/ii.conf 構成ファイルの ii_bitmap 変数によって、次のようにビットマップボリュームの動作セマンティクスが決定されます。

0 ビットマップがメモリー内のみ、または復元再開操作で保持されることを示します。

1 ビットマップがメモリー内とディスク上に保持されることを示します。これがデフォルト値です。

ii_bitmap=0 の使用中にシステム障害が発生すると、シャドウボリュームの整合性がくずれ、高速再同期が不可能となることがあります。

ポイントインタイムコピーをネットワークストレージのコンポーネントである遠隔ミラーと組み合わせて使用する場合や Sun Cluster 内で使用する場合には、`ii_bitmap=1` を設定します。

`/usr/kernel/drv/ii.conf` 構成ファイルの `ii_debug` 変数によって、ポイントインタイムコピーの処理中にシステムコンソール `/dev/console` に出力される情報ロギングの量が決まります。

`0` ロギングをシステムコンソールに送信しないことを示します。

`1` 情報ロギングをシステムコンソールに送信することを示します。

`2` 開発ロギングをシステムコンソールに送信することを示します。

オプション

`iiadm` ユーティリティーは次のオプションをサポートします。

`-e{ind|dep}` *master_vol shadow_vol bitmap_vol*

指定されたマスター、シャドウ、ビットマップの各ボリュームでポイントインタイムコピーを有効化します。

シャドウセット有効化処理は、指定されたボリュームにアクセスできること、*shadow_vol* がマウントされていないこと、および *bitmap_vol* のサイズが作成するシャドウセットのタイプに適したものであることを確認します。さらにこの処理は、それらのボリュームが SV ドライバの制御下にあることを確認し (制御下にないときは制御下に入れる)、ビットマップボリュームを初期化します。そして、ボリュームセットが独立シャドウセットである場合には、フルコピー操作が開始されます。

有効化が成功すると、ポイントインタイムコピーは指定された *master_vol*、*shadow_vol*、*bitmap_vol* の名前と有効化タイプ (`ind` または `dep`) をポイントインタイムコピー構成ストア内に格納します。構成ストアには、現在構成されているすべてのポイントインタイムコピーボリュームセットと関連する構成属性が含まれています。(独立および依存シャドウボリュームのセマンティクスに関する前述の説明を参照。)

master_vol は、ポイントインタイムスナップショットの作成元となるボリュームです。

shadow_vol は、ポイントインタイムスナップショットが格納されるボリュームです。

bitmap_vol は、シャドウボリュームとマスターボリュームの差異を追跡するために使用されます。ポイントインタイムコピーのシャドウ操作が中断または再開された場合、(カーネルメモリーに保持されている) ビットマップボリュームを永続記録領域に格納したりそこから取得したりできます。ビットマップボリュームに関連付けられた記憶領域には、シャドウボリュームの記憶領域と同等の冗長性を持たせるべきです。

shadow_vol の名前は、*volume_set* 名を指定する必要のあるすべての `iiadm` オプションでのポイントインタイムコピーシャドウセットの名前になります。

`-d` *volume_set*
指定された *volume_set* に関連付けられたポイントインタイムコピーボリュームセットを無効化します。

ポイントインタイムコピーの動作モードが前述の `-e ind` オプションで指定される独立モードであった場合、シャドウボリュームのデータには、無効化する前と同じデータが含まれています (書き込みが一切発生しなかった場合)。マスターボリュームとシャドウボリュームはこの時点でスタンドアロンのポイントインタイムコピーになっているため、ユーザーはそれらのボリュームにアクセスできます。

フルコピーがアクティブになっている間、独立ボリュームはまるで依存ボリュームであるかのように動作します。ボリュームのモードがフルコピーでないことを保証するには、次のコマンドを発行してフルコピーが完了するまで待機します。

```
# iiadm -w volume_set
```

[`-p`] `-u s` *volume_set*
マスターボリュームからシャドウボリュームを更新します。

マスターボリュームのポイントインタイムコピーをシャドウボリュームに更新します。*volume_set* は、マスターボリュームとシャドウボリュームを含むポイントインタイムコピーシャドウセットです。このオプションは、シャドウボリュームの高速再同期を提供し、マスターの増分コピーを作成します。この更新では、マスターボリュームとシャドウボリュームとで異なるというフラグの付いた 32K バイトセグメントがすべてコピーされます。コピーされるのは、マスターボリュームのすべてのデータではなく、変更されたデータだけです。データがコピーされている間、シャドウはマスターボリュームに依存します。

このオプションを使用する前に、ボリュームへのワークロードを短時間停止します。ホストアプリケーションがボリュームに書き込みを行わないようにします。これにより、ポイントインタイムデータの一貫性が保証されます。対象ボリュームをほかの操作で使用する前に、このコピー操作や更新操作のステータスを、`iiadm -i` *volume_set* で視覚的に確認することも、(シェルやスクリプト経由で) `iiadm -w` *volume_set* を使用して対話形式で確認することもできます。

このコマンドは、`iiadm -p -u s` のようにオプション `-p` を使用することで、PID (Process IDentifier) のロックをサポートします。このオプションを有効にすると、ほかのプロセスの新しいポイントインタイムスナップショットの取得によって以前のポイントインタイムデータが無効になるのを防ぐことができます。

[`-p`] [`-n`] `-u m` *volume_set*
マスターボリュームのポイントインタイムコピーをシャドウから更新します。*volume_set* は、マスターおよびシャドウを含むポイントインタイムコピーボ

リュームセットです。このオプションは、マスターボリュームの高速再同期を提供し、シャドウの増分コピーを作成します。この更新では、マスターボリュームとシャドウボリュームとで異なることを示すフラグの付いた 32K バイトセグメントがすべてコピーされます。コピーされるのは、シャドウボリュームのすべてのデータではなく、変更されたデータだけです。データがコピーされている間、マスターはシャドウボリュームに依存します。

このオプションを使用する前に、ボリュームへのワークロードを短時間停止します。ホストアプリケーションがボリュームに書き込みを行わないようにします。これにより、ポイントインタイムデータの一貫性が保証されます。対象ボリュームをほかの操作で使用する前に、このコピー操作や更新操作のステータスを、`iiadm -i` *volume_set* で視覚的に確認することも、(シェルやスクリプト経由で) `iiadm -w` *volume_set* を使用して対話形式で確認することもできます。

このコマンドでは、マスターボリュームのデータを間違って上書きしてしまわないように、クエリーが有効化されます。このコマンドオプションをスクリプト内で使用する場合は、`-n` オプションを追加してクエリーが発生しないようにします。

このコマンドは、`iiadm -p -u m` のようにオプション `-p` を使用することで、PID (Process IDentifier) のロックをサポートします。このオプションを有効にすると、ほかのプロセスの新しいポイントインタイムスナップショットの取得によって以前のポイントインタイムデータが無効になるのを防ぐことができます。

[`-p`] `-c s` *volume_set*

マスターボリュームをシャドウにコピーします。

マスターボリュームのポイントインタイムコピーをシャドウボリュームに作成します。*volume_set* は、マスターおよびシャドウを含むポイントインタイムコピーボリュームセットです。このオプションは、マスターボリュームのポイントインタイムコピー内のすべてのデータをシャドウボリュームに書き込みます。データがマスターからシャドウにコピーされている間、シャドウはマスターボリュームに依存します。

このオプションはフルボリュームコピーを実行します。独立シャドウボリュームのデータの整合性が疑われる場合以外は、`iiadm -u s` を使用してください。それ以外の場合は、このオプションを使用してマスターボリュームとシャドウボリュームの同期をとる、つまりそれぞれのボリュームのデータを一致させます。

このオプションを使用する前に、ボリュームへのワークロードを短時間停止します。ホストアプリケーションがボリュームに書き込みを行わないようにします。これにより、ポイントインタイムデータの一貫性が保証されます。対象ボリュームをほかの操作で使用する前に、このコピー操作や更新操作のステータスを、`iiadm -i` *volume_set* で視覚的に確認することも、(シェルやスクリプト経由で) `iiadm -w` *volume_set* を使用して対話形式で確認することもできます。

このコマンドは、iiadm -p -c s のようにオプション -p を使用することで、PID (Process IDentifier) のロックをサポートします。このオプションを有効にすると、ほかのプロセスの新しいポイントインタイムスナップショットの取得によって以前のポイントインタイムデータが無効になるのを防ぐことができます。

-c m *volume_set*
シャドウボリュームをマスターにコピーします。

シャドウボリュームのポイントインタイムコピーをマスターボリュームに作成します。*volume_set* は、マスターおよびシャドウボリュームを含むポイントインタイムコピーボリュームセットです。このオプションは、シャドウボリュームのポイントインタイムコピー内のすべてのデータをマスターボリュームに書き込みます。データがシャドウからマスターにコピーされている間、マスターはシャドウボリュームに依存します。

このオプションはフルボリュームコピーを実行します。独立マスターのデータの整合性が疑われる場合以外は、iiadm -u m を使用してください。それ以外の場合は、このオプションを使用してマスターボリュームとシャドウボリュームの同期をとる、つまりそれぞれのボリュームのデータを一致させます。

このオプションを使用する前に、ボリュームへのワークロードを短時間停止します。ホストアプリケーションがボリュームに書き込みを行わないようにします。これにより、ポイントインタイムデータの一貫性が保証されます。対象ボリュームをほかの操作で使用する前に、このコピー操作や更新操作のステータスを、iiadm -i *volume_set* で視覚的に確認することも、(シェルやスクリプト経由で) iiadm -w *volume_set* を使用して対話形式で確認することもできます。

このコマンドでは、マスターボリュームのデータを間違って上書きしてしまわないように、クエリーが有効化されます。このコマンドオプションをスクリプト内で使用する場合は、-n オプションを追加してクエリーが発生しないようにします。

このコマンドは、iiadm -p -c m のようにオプション -p を使用することで、PID (Process IDentifier) のロックをサポートします。このオプションを有効にすると、ほかのプロセスの新しいポイントインタイムスナップショットの取得によって以前のポイントインタイムデータが無効になるのを防ぐことができます。

-a *volume_set*
マスターボリュームとシャドウボリュームとの間で現在アクティブになっている可能性のあるコピー操作をすべて中止します。*volume_set* は、マスターおよびシャドウボリュームを含むポイントインタイムコピーボリュームセットです。iiadm -a の実行後、対象 (マスターまたはシャドウ) ボリュームに対する更新またはコピーは不完全になっています。この時点で、コピー先のボリュームはコピー元のボリュームの依存コピーになっています。更新またはコピーコマンドオプションを再発行してボリュームの再同期を行います。

`[-p] [-n] -w` *volume_set*

実行中のすべてのコピー操作または更新操作が完了するか異常終了するまで待機します。*volume_set* は、マスターおよびシャドウボリュームを含むポイントインタイムコピーボリュームセットです。

このオプションは、現在のポイントインタイムコピー操作が完了するまで待機するため、後続の `iiadm` コマンドが (シェルやスクリプトから) 実行されるのを防ぎます。このコマンドオプションは、コピー操作や更新操作が完了したことを確認する必要がある場合に使用します。

このコマンドは PID (Process IDentifier) のロック解除をサポートします。コマンド `iiadm -p {-c|-u} {m|s}` を使用して前に行われたコピーや更新で `-p` オプションが指定されていて、現在の PID がポイントインタイムデータをロックした PID であった場合、待機処理の完了時にこのオプションはそのデータのロックを解除します。

`-i` *volume_set*

現在有効化または中断されているポイントインタイムコピーボリュームセットのステータスを表示します。*volume_set* は、マスターおよびシャドウボリュームを含むポイントインタイムコピーボリュームセットです。*volume_set* を指定しなかった場合は、構成されているすべてのポイントインタイムコピーボリュームセットのステータスが表示されます。

`-l`

現在構成されているすべてのポイントインタイムコピーボリュームを一覧表示します。

`-O` *overflow_vol*

このオプションを使用すると、指定した *overflow_vol* がポイントインタイムコピーによって初期化され、あとで小型依存シャドウボリュームと組み合わせてオーバーフローボリュームとして使用できるようになります。このオーバーフローボリュームの効率的な共有使用を促進するため、必要に応じてこのボリュームを複数のポイントインタイムコピーボリュームセットに関連付けられるようになっています。

overflow_vol の初期化中、このオプションの起動元は次の質問に答える必要があります。「Initialize this overflow volume?yes/no」続行するには、「yes」、「no」のいずれかの応答が必要です。

このオプションでは `-n` オプションがサポートされており、要求された処理がプロンプトなしで実行できます。このオプションは、スクリプト内に含める場合に便利です。`-n` オプションを最初に指定する必要があります。たとえば、「`iiadm -nO vol`」は有効ですが、「`iiadm -On vol`」は無効です。

特に `-n` オプションを使用する場合には、指定された *overflow_vol* のデータを初期化してよいか確認してください。

-A *overflow_vol volume_set*
このオプションは、指定された *overflow_vol* を有効化し、あとで小型依存シャドウボリュームのサイズが小さすぎた場合やボリュームセットに対して予定されていない量の変更が発生した場合にオーバーフローボリュームとして使用できるようにします。オーバーフローした変更データは、関連付けられたオーバーフローボリュームにリダイレクトされます。*volume_set* は、マスターおよびシャドウボリュームを含むポイントインタイムコピーボリュームセットです。

overflow_vol が初期化されていなかった場合、このオプションは、*overflow_vol* を初期化 (-O オプションを参照) したあと、*overflow_vol* を *volume_set* に接続します。

overflow_vol が事前に初期化されていた場合、このオプションは、*overflow_vol* を *volume_set* に接続します。

このオプションでは -n オプションがサポートされていますが、このオプションを使用すると、要求された処理がプロンプトなしで実行されます。このオプションは、スクリプト内に含める場合に便利です。-n オプションを最初に指定する必要があります。たとえば、「iiadm -nA vol」は有効ですが、「iiadm -An vol」は無効です。

特に -n オプションを使用する場合には、指定された *overflow_vol* のデータを初期化してよいか確認してください。

-D *volume_set*
このオプションは、指定された *volume_set* に現在関連付けられているオーバーフローボリュームを削除します。オーバーフローボリュームが現在 *volume_set* によって使用されている場合、この操作は失敗し、「Overflow volume still in use」というエラーメッセージが表示されます。この状況を解決するには、次に説明する操作のいずれかを *volume_set* に対して実行します。これらの操作は、このボリュームセットに関連付けられたすべてのオーバーフロー書き込みを短時間でクリアします。

abort(-a)
コピー操作を中止します。

disable(-d)
ボリュームセットを破棄します。

update(-u)
ボリュームセットを更新します。

-L
このオプションは、1 つ以上のボリュームセットに関連付けられたオーバーフローボリュームをすべて一覧表示します。

-Q *overflow_vol*
このオプションは、*overflow_vol* の現在のステータスを表示します。

-E *volume_set*
volume_set で指定されたポイントインタイムコピーボリュームセットの独立シャドウボリュームをエクスポートします。このシャドウボリュームは、マルチポートデバイスなどの実現技術を利用して、別のホストから読み取り/書き込み可能になるものです。差異ビットマップの管理はこの別のホストが担当することになりますが、このビットマップは、シャドウボリュームをマスターボリュームに再結合する際に、ローカルで記録されたマスターとの差異とマージするために使用されます。シャドウボリュームのエクスポート中は、そのボリュームに対して更新またはコピー操作を行わないでください。エクスポートコマンドを呼び出す前に、`iiadm -w` *volume_set* コマンドを実行します。

-I *volume_set bitmap_vol*
volume_set で指定されたポイントインタイムコピーボリュームセットの独立シャドウボリュームをインポートします。このシャドウボリュームは、マルチポートデバイスなどの実現技術を使用して事前にあるホストからエクスポートされたものである必要があります。インポート操作を行うと、このホストが、ボリュームの変更に伴う差異ビットマップの維持管理を開始します。*bitmap_vol* は、シャドウボリュームが最初にシャドウグループ内に形成されたときに使用されたビットマップと同じにならないようにしてください。

エクスポート/インポートされた独立シャドウボリュームがこの別のノードで不要になったら、*bitmap_vol* とその関連 *shadow_vol* が一貫性を保てるように、結合操作の実行前に無効化コマンドを入力する必要があります。たとえば、

```
# iiadm -d volume_set
```

-J *volume_set bitmap_vol*
bitmap_vol を使用して、*volume_set* をポイントインタイムコピーボリュームセットのマスターボリュームセットに結合します。設定したビットマップボリュームが読み取られて元のボリュームとマージされ、マスター、シャドウ、ビットマップの各ボリュームから構成される、元のボリュームセットが再構築されます。マージ対象となる *bitmap_vol* は、独立シャドウボリュームをインポートしたノード上で取得されたものです。インポート先のマシン上では、ビットマップがコピーされてからシャドウが再度インポートされるまでの間、シャドウボリュームに対する書き込みが発生してはいけません。

-g *group_name* -m *volume_set [volume_set2 ...]*
1 つ以上の既存ポイントインタイムコピー *volume_set* をユーザー指定の *group_name* に追加します。このように 1 つ以上のポイントインタイムコピーボリュームセットをグループ内で関連付けることで、*group_name* 内のすべてのボリュームセット全体に対し、後述の一連の `iiadm` オプションを実行できるようになります。

グループ内のすべてのポイントインタイムコピーセットにわたって原子的に実行できるコマンドは、`COPY` (`-c`) と `UPDATE` (`-u`) だけです。その他のグループ化された `iiadm` コマンドはすべて、グループの各メンバーに対して順番に実行されます。

iiadm グループコマンドの構文は次のとおりです。

iiadm -g *group_name* [*options*]

オプションは、次のとおりです。

-a
group_name 内のすべてのセットでコピー操作を中止します。

-A
group_name 内のすべてのセットに *overflow_vol* を接続します。

-c {s | m}
group_name 内のすべてのセットでシャドウ/マスターをコピーします。

-D
group_name 内のすべてのセットから *overflow_vol* を切り離します。

-d
group_name 内のすべてのセットを無効化します。

-E
group_name 内のすべてのボリュームセットをエクスポートします。

-i
group_name 内のすべてのボリュームセットのステータス。

-l
group_name 内のすべてのボリュームセットを一覧表示します。

-L
すべてのグループを一覧表示します。

-n
マスターボリュームを本当に更新するかどうかをユーザーに確認しません。

-P
group_name 内のすべてのボリュームセットにパラメータを設定します。

-R
group_name 内のすべてのボリュームセットをリセットします。

-u {s | m}
group_name 内のすべてのセットでシャドウ/マスターを更新します。

-w
group_name 内のすべてのボリュームセット上で待機します。

-g -" " -m *volume_set* [*volume_set2* ...]
1 つ以上の既存ポイントインタイムコピー *volume_set* を、現在関連付けられている *group_name* から削除します。すべてのポイントインタイムコピー *volume_set* はデフォルトで、またはユーザー指定の *group_name* に移動されるまで、空 (" ")

のグループ内に存在しています。この関連付けを行うと、空 (" ") の `iiadm` *group_name* に対して前述のすべての `iiadm` グループコマンドを実行できるようになります。

`-C` *cluster_tag*
このポイントインタイムコピーオプションは、Sun Cluster のリソースグループまたはディスクグループに属するボリュームだけに構成操作を制限する修飾子です。

Sun Cluster に対応したボリュームマネージャーを持つ Sun Cluster では `iiadm` が正しいディスクグループ情報を自動的に取得するため、このオプションは通常、ボリュームが広範囲をカバーするリソースグループの一部になっていないかぎり不要です。

ローカルノードでしかボリュームにアクセスできないような Sun Cluster では、Sun Cluster のリソースグループまたはディスクグループの一部になっていないボリュームを示すために、特殊な *cluster_tag* `local` が使用されます。

cluster_tag 引数として `-L` が指定された場合、`iiadm` は、ポイントインタイムコピーに関連するクラスタタグをすべて一覧表示します。

Sun Cluster パッケージがインストールまたは構成されていない Solaris システム上では、このオプションは無効になります。

`-h`
`iiadm` の使用法の要約を出力します。

`-v`
ポイントインタイムコピーソフトウェアコンポーネントの現在のバージョンを表示します。

この節の残りのコマンドの使用方法について支援が必要な場合には、Sun Enterprise サービス窓口までご連絡ください。

`-P` *delay unit volume_set*
指定された *volume_set* のポイントインタイムコピーボリュームセットチューニングパラメータを *unit* 件の入出力ごとの *delay* クロック刻み数に変更します。delay の範囲は 2 から 10000、unit の範囲は 100 から 60000 であり、どちらも両端を含みます。

`-R` *volume*
あるボリュームで障害が発生すると、ポイントインタイムコピーはそのボリュームをオフライン状態にします。ボリュームの交換が完了したら、このオプションを使用してボリュームをオンライン状態に戻します。ポイントインタイムコピーボリュームセット内の関連する依存ボリュームもオンライン状態になります。ボリュームをオンライン状態にしたあと、このコマンドはさらに、必要なすべてのポイントインタイムボリューム更新を開始します。

終了ステータス

`0` コマンドが正常に完了しました。

`>0` エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-point-in-time-copy
インタフェースの安定性	確実

関連項目

`dscfg(1M)`, `svadm(1M)`, `ds.log(4)`, `rdc.cf(4)`, `attributes(5)`, `ii(7D)`, `sv(7D)`

名前

iicpbmp – Availability Suite ポイントインタイムビットマップボリュームのコピー

形式

`iicpbmp [-c] old_bitmap new_bitmap...`

機能説明

`iicpbmp` コマンドは、Availability Suite ポイントインタイムのビットマップボリュームをコピーし、新しいビットマップ名との整合性がとれるようにビットマップヘッダーを書き換えます。シャドウセットの構成エントリが書き換えられ、新しいビットマップの場所が反映されます。

古いビットマップボリュームや新しいビットマップボリュームの現在の使用状況はチェックされません。`iicpbmp` コマンドを実行するのは、古いビットマップを使用するポイントインタイムコピーシャドウセットが中断されている場合だけにしてください。

オプション

`-c` このビットマップを使用するポイントインタイムシャドウセットの Availability Suite 構成の更新を試みません。このオプションは、ビットマップの複製を生成しますが、古いビットマップボリュームを使用するシャドウセットには影響を与えません。

オペランド

old_bitmap new_bitmap
ポイントインタイムの古いビットマップボリュームと新しいビットマップボリューム。

警告

`iicpbmp` を実行するのは、システムがシングルユーザーモードの場合だけにしてください。`iicpbmp` は、コピーを行う時点でポイントインタイムコピーセットが使用中かどうかのチェックを試みません。ポイントインタイムコピーがシャドウセットを使用している最中に `-c` フラグなしで `iicpbmp` を実行すると、ポイントインタイムコピーの次回起動時にシャドウセットの不整合が発生します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-point-in-time-copy
インタフェースの安定性	確実

関連項目

`iiadm(1M)`, `iicpshd(1M)`, `attributes(5)`

名前

iicpshd – Availability Suite ポイントインタイムシャドウボリュームのコピー

形式

`iicpshd [-s] old_shadow new_shadow...`

機能説明

`iicpshd` コマンドは、Availability Suite Instant Image シャドウボリュームをコピーし、ビットマップヘッダーと Availability Suite 構成を更新して新しいシャドウボリュームの情報を反映させます。

古いシャドウボリュームや新しいシャドウボリュームの現在の使用状況はチェックされません。`iicpshd` コマンドを実行するのは、古いシャドウボリュームを使用する Instant Image シャドウセットが中断されている場合だけにしてください。

オプション

`iicpshd` コマンドは、次のオプションをサポートします。

`-s` ポイントインタイムシャドウセットの StorageTek 構成情報は更新しますが、古いシャドウボリュームから新しいシャドウボリュームへのデータコピーは行いません。

オペランド

`iicpshd` コマンド行のオペランドは、次のとおりです。

old_shadow new_shadow
`iicpshd` は、古い Availability Suite Instant Image シャドウボリュームのデータを新しいシャドウボリュームにコピーし、ビットマップヘッダーと構成データを更新します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-point-in-time-copy
インタフェースの安定性	確実

関連項目

`iiadm(1M)`, `iicpbmp(1M)`, `attributes(5)`

警告

`iicpshd` を実行するのは、システムがシングルユーザーモードの場合だけにしてください。ユーザーが `iicpshd` を実行した際に、このコマンドは、ポイントインタイムセットが使用中かどうかのチェックを試みません。ポイントインタイムが古いシャドウボリュームを使用している最中に `-s` フラグを指定して `iicpshd` を実行すると、シャドウボリュームのデータが失われる可能性があります。`-s` オプションを使用する場合には、古いシャドウボリュームのデータを新しいシャドウボリュームに手動でコピーする必要があります。

名前

ikeadm – インターネット鍵交換 (IKE) のパラメータおよび状態の操作

形式

```
ikeadm [-np]

ikeadm [-np] get [debug | priv | stats | defaults]

ikeadm [-np] set [debug | priv] [level] [file]

ikeadm [-np] [get | del] [p1 | rule | preshared] [id]

ikeadm [-np] add [rule | preshared] { description }

ikeadm [-np] token [login | logout] PKCS#11_Token_Object

ikeadm [-np] [read | write] [rule | preshared | certcache] file

ikeadm [-np] dump [p1 | rule | preshared | certcache | groups
     | encralgs | authalgs]

ikeadm [-np] flush [p1 | certcache]

ikeadm help
     [get | set | add | del | read | write | dump | flush | token]
```

機能説明

ikeadm ユーティリティーは、インターネット鍵交換 (IKE) プロトコルデーモン in.iked(1M) から情報を取得したり、その構成を操作したりします。

ikeadm は、サポートされる 1 つ以上のオブジェクトタイプに対して実行可能な一連の操作をサポートしています。ikeadm は、引数なしで呼び出されると対話型モードに入り、標準出力にプロンプトを出力したあと、ファイルの終わりに達するまで標準入力からコマンドを受け入れます。

ikeadm は機密性の高いキーイング情報を操作するため、このコマンドを使用するにはスーパーユーザーである必要があります。さらに、使用可能なコマンドの一部では、デーモンが特権モードで動作している必要がありますが、その動作が確立されるのは、デーモンの起動時です。

このコマンドを安全に使用する方法の詳細については、セキュリティーを参照してください。

オプション

次のオプションがサポートされています。

-n
: アクションの報告時に、ホスト名やネットワーク名をシンボルで出力できないようにします。これは、すべてのネームサーバーが停止したか、またはその他の理由で到達不能になった場合などに役立ちます。

-p
: パラノイド。セキュリティーアソシエーションを保存する場合でもキーイング材料を一切出力しません。このフラグが有効になると、実際の 16 進数値の代わりに x を出力します。

使用法

コマンド　次のコマンドがサポートされています。

`add`
指定されたオブジェクトを追加します。このオプションを使用すると、現在の(実行中の) in.iked の構成に新しいポリシールールや事前共有鍵を追加できます。新しい事前共有鍵を追加する場合、キーイング材料がコマンドに含まれるため、コマンド行からコマンドを呼び出すことはできません。追加するルールや鍵を指定するには、「ID 形式」節で説明する適切な ID-値ペアを使用します。

`del`
`in.iked` の現在の構成から、特定の 1 つまたは複数のオブジェクトを削除します。この操作は、IKE (第 1 段階) SA、ポリシールール、および事前共有鍵で使用できます。削除するオブジェクトの指定方法については、「ID 形式」を参照してください。

`dump`
`in.iked` が認識しているタイプのうち、指定されたタイプのオブジェクトをすべて表示します。このオプションを使用すると、第 1 段階 SA、ポリシールール、事前共有鍵、実装された Diffie-Helman グループ、第 1 段階で使用可能な暗号化アルゴリズムと認証アルゴリズム、証明書キャッシュのすべてを表示できます。このコマンドでは大量の出力が生成される可能性があります。

`flush`
`in.iked` から IKE (第 1 段階) SA またはキャッシュ内の証明書をすべて削除します。

`certcache` をフラッシュすると、(その副作用として) 追加または削除された新しい証明書がすべて IKE に更新されます。

`get`
指定されたオブジェクトを検索して表示します。現在のデバッグレベルまたは特権レベル、デーモンのグローバル統計やデフォルト値、特定の IKE (第 1 段階) SA、特定のポリシールール、または特定の事前共有鍵を表示するために使用できます。最後の 3 つのオブジェクトタイプでは識別情報を渡す必要があります。各オブジェクトタイプの適切な指定方法については後述します。

`help`
コマンドの簡単な要約を出力します。ただし、特定のコマンドが指定された場合は、そのコマンドに関する情報を出力します。

`read`
`in.iked` の現在の構成を更新するために、デフォルトの場所または指定されたファイルのいずれかからポリシールールまたは事前共有鍵を読み取ります。

`set`
現在のデバッグレベルまたは特権レベルを調整します。デバッグレベルを変更する場合、出力ファイルを指定することもできます。デーモンがバックグラウンド

で動作しており、ファイルへの出力を現在行なっていない場合は、出力ファイルを指定する必要があります。特権レベルを変更する場合は、アクセスレベルを下げるための調整しか行うことができません。ikeadm を使用してレベルを上げることはできません。

`write`
: `in.iked` の現在のポリシールールセットまたは事前共有鍵セットを指定されたファイルに書き込みます。書き込み先のファイルを指定する必要があります。このコマンドは、既存の構成ファイルを上書きするために使用するべきではありません。

`token`
: PKCS#11 トークンオブジェクトにログインしてキーイング材料へのアクセスを許可するか、ログアウトしてキーイング材料へのアクセスを無効にします。

  `token` は、次の承認を持つ通常ユーザーとして実行できます。

  - `token` ログイン: `solaris.network.ipsec.ike.token.login`
  - `token` ログアウト: `solaris.network.ipsec.ike.token.logout`

オブジェクトタイプ

debug
: デーモンのデバッグレベルを指定します。これにより、デーモンから提供される、操作に関する出力の量や種類が決まります。デバッグレベルは実際にはビットマスクであり、その個々のビットによってそれぞれ異なるタイプの情報が有効化されます。

機能説明	フラグ	ニックネーム
証明書管理	0x0001	cert
鍵管理	0x0002	key
操作可能	0x0004	op
第 1 段階 SA の作成	0x0008	phase1
第 2 段階 SA の作成	0x0010	phase2
PF_KEY インタフェース	0x0020	pfkey
ポリシー管理	0x0040	policy
指定構築	0x0080	prop
ドアインタフェース	0x0100	door
構成ファイルの処理	0x0200	config
ラベルの処理	0x0400	label
すべてのデバッグフラグ	0x07ff	all

デバッグレベルを指定する場合、数値 (10 進または 16 進)、ニックネーム文字列のどちらを指定してもかまいません。たとえば、88、0x58、phase1+phase2+policy はすべて同等であり、第 1 段階 SA 作成、第 2 段階 SA 作成、およびポリシー管理のデバッグを有効にします。ニックネーム文字列を使用して特定タイプの情報を除外することもできます。all-op は、操作メッセージ以外のすべてのデバッグを有効にする効果を持ちますが、これは数値 1019 や 0x3fb と同等になります。

priv

デーモンのアクセス特権レベルを指定します。指定可能な値は次のとおりです。

```
Description                  Level   Nickname
Base level                   0       base
Access to preshared key info 1       modkeys
Access to keying material    2       keymat
```

デフォルトでは、in.iked はベースレベルで起動されます。コマンド行オプションを使用すると、より高いレベルでデーモンを起動できます。ikeadm では、レベルを下げることはできますが、レベルを上げることはできません。

目的の特権レベルを指定する場合は、数値レベルとニックネームのどちらを使用してもかまいません。

事前共有鍵の取得、追加、削除、ダンプ、読み取り、または書き込みを行うには、少なくとも事前共有鍵情報へのアクセスを許可する特権レベルが必要になります。ただし、(get または dump のいずれかのコマンドを使用して) 事前共有鍵を表示する際に鍵自体が使用可能になるのは、特権レベルがキーイング材料へのアクセスを許可する場合だけです。第 1 段階 SA を表示する場合も、これと同様です。

stats

デーモンのグローバル統計であり、第 1 段階 SA 作成の成功と失敗の両方をカバーします。

報告される統計情報には次が含まれます。

- ローカルエンティティーが開始した P1 SA の現在の数
- ローカルエンティティーがレスポンダになった P1 SA の現在の数
- 起動以降にローカルエンティティーが開始した P1 SA の合計数
- 起動以降にローカルエンティティーがレスポンダになった P1 SA の合計数
- 起動以降にローカルエンティティーがイニシエータになった試行 P1 SA の合計数。失敗した試行も含まれます。
- 起動以降にローカルエンティティーがレスポンダになった試行 P1 SA の合計数。失敗した試行も含まれます。
- P1 SA を開始しようとして失敗した合計回数 (ピアが応答しなかったためにエラーが発生した場合)

- P1 SA を開始しようとして失敗した合計回数 (ピアが応答した場合)
- ピアがイニシエータになった失敗 P1 SA の合計数
- PKCS#11 ライブラリが使用中かどうかと、使用中の場合はその読み込まれた PKCS#11 ライブラリ例 11を参照してください。

defaults
: `in.iked` デーモンで使用されるデフォルト値を表示します。値の中には、デーモンの構成ファイルで上書きできるものもあります (`ike.config(4)` を参照)。それらの値では、`get defaults` の出力時にトークン名が表示されます。出力には、構成トークンがデフォルトを変更した箇所が反映されます。

  デフォルト値は、ピアシステムから有効な代替案が提出された場合は無視される可能性があります。また、`ike.config` で確立されたルール単位の値で上書きされる可能性もあります。そのような場合も、`get defaults` コマンドは、デフォルトの上書きに使用された値ではなく、デフォルト値を表示し続けます。

p1
: IKE 第 1 段階 SA。`p1` オブジェクトは IP アドレスペアまたは cookie ペアのいずれかで識別されますが、その識別形式については後述します。

rule
: IKE ポリシールール。指定されたローカルアイデンティティーとリモートアイデンティティーとの間の第 1 段階 SA で使用可能なセキュリティー特性を定義します。ルールはラベルで識別されますが、その識別形式については後述します。

preshared
: 事前共有鍵。ローカルとリモートの識別、および適用可能な IKE モードも含まれます。事前共有鍵は IP アドレスペア、アイデンティティーペアのいずれかで識別されますが、その識別形式については後述します。

ID 形式

`add`、`del`、`get` などのコマンドでは、コマンド行に追加情報を指定する必要があります。削除コマンドと取得コマンドで必要になるのは、特定のオブジェクトを識別する最小限の情報だけです。追加コマンドでは完全なオブジェクトを指定する必要があります。

最小限の識別はほとんどの場合、値のペアによって実現されます。IP アドレスの場合、ローカルアドレスのあとにリモートアドレスを指定しますが、その形式は IPv4 アドレス用のドット表記、IPv6 アドレス用のコロン区切り 16 進数形式、またはホスト名データベース内に存在するホスト名のいずれかになります。複数のアドレスに展開されるホスト名を指定した場合、アドレスの可能な各組み合わせについて、要求された操作が1 度に複数回実行されます。

アイデンティティーペアは、ローカルのタイプ-値ペアとそれに続くリモートのタイプ-値ペアとで構成されます。有効なタイプは次のとおりです。

prefix
: アドレスの接頭辞。

fqdn
完全修飾ドメイン名。

domain
ドメイン名。fqdn の同義語です。

user_fqdn
user@fqdn という形式のユーザーアイデンティティー。

mailbox
`user_fqdn` の同義語。

cookie ペアは、第 1 段階セキュリティーアソシエーション (SA) の作成時にその SA に割り当てられた 2 つの cookie から構成され、最初の cookie はイニシエータ用、そのあとの cookie はレスポンダ用です。cookie は 64 ビットの数値です。

最後に、(ポリシールールの識別子として使用される) ラベルは、ルール作成時にルールに割り当てられる文字列です。

追加コマンドでのルールや事前共有鍵の記述形式は、in.iked 構成ファイルの形式規則に従います。どちらも、中括弧 ({ と }) で囲まれた一連の ID-値ペアから構成されます。ルールや事前共有鍵の形式の詳細については、`ike.config(4)` と `ike.preshared(4)` を参照してください。

セキュリティー

`ikeadm` コマンドを使用すると、特権ユーザーが暗号化キーイング情報を入力できます。そのような情報に悪意のあるユーザーがアクセスできると、IPsec トラフィックのセキュリティーが損なわれます。`ikeadm` コマンドを使用する際には、次の問題を考慮するようにしてください。

- TTY の情報がネットワーク上を流れるか (対話モード)

  そうである場合、キーイング材料のセキュリティーは、この TTY のトラフィックに対するネットワークパスのセキュリティーになります。平文の telnet または rlogin セッション経由で `ikeadm` を使用するのは危険です。ウィンドウイベントを読み取ることのできる隠密プログラムが存在する場合、ローカルウィンドウでもその攻撃には無防備となります。

- ファイルがネットワーク経由でアクセスされたり、外部から読み取り可能であったりするか (読み取り/書き込みコマンド)

  ネットワーク上にマウントされたファイルを読み取る際に、悪意のあるユーザーがその情報を盗む可能性があります。外部から読み取り可能なファイルにキーイング材料が格納されている場合も危険です。

送信元アドレスがネットワーク経由で検索可能なホストである場合に、ネームシステム自体の安全性が損なわれると、使用されているすべての名前が信用できなくなります。

セキュリティーの脆弱性はしばしば、ツール自体に原因があるのではなく、ツールの間違った適用方法に原因があります。管理者には、`ikeadm` コマンドの使用時に十

分な注意を払うことをお勧めします。もっとも安全な操作モードはおそらく、コンソール上やその他の物理的に接続された TTY 上で作業を行うことです。

このテーマの追加情報については、Bruce Schneier 著『Applied Cryptography: Protocols, Algorithms, and Source Code in C』に含まれる Matt Blaze 氏の後書きを参照してください。

使用例

例 1 すべての第 1 段階セキュリティーアソシエーションの削除

次のコマンドは、すべての第 1 段階セキュリティーアソシエーションを削除します。

```
example# ikeadm flush p1
```

例 2 すべての第 1 段階セキュリティーアソシエーションの表示

次のコマンドは、すべての第 1 段階セキュリティーアソシエーションを表示します。

```
example# ikeadm dump p1
```

例 3 特定の第 1 段階セキュリティーアソシエーションの削除

次のコマンドは、指定された第 1 段階セキュリティーアソシエーションを削除します。

```
example# ikeadm del p1 local_ip remote_ip
```

例 4 ファイルからのルールの追加

次のコマンドはファイルからルールを追加します。

```
example# ikeadm add rule rule_file
```

例 5 事前共有鍵の追加

次のコマンドは事前共有鍵を追加します。

```
example# ikeadm
     ikeadm> add preshared { localidtype ip localid local_ip
             remoteidtype ip remoteid remote_ip ike_mode main
             key 1234567890abcdef1234567890abcdef }
```

例 6 すべての事前共有鍵のファイルへの保存

次のコマンドは、すべての事前共有鍵をファイルに保存します。

```
example# ikeadm write preshared target_file
```

例 7 特定のルールの表示

次のコマンドは特定のルールを表示します。

```
example# ikeadm get rule rule_label
```

例 8　ike.config からの新しいルールの読み込み

次のコマンドは、ike.config ファイルから新しいルールを読み込みます。

```
example# ikeadm read rules
```

例 9　特権レベルの引き下げ

次のコマンドは特権レベルを引き下げます。

```
example# ikeadm set priv base
```

例 10　デバッグレベルの表示

次のコマンドは現在のデバッグレベルを表示します

```
example# ikeadm get debug
```

例 11　stats を使用したハードウェアアクセラレータの確認

次の例は、統計情報の末尾に省略可能な行が含まれる様子を示したものです。この行は該当する場合に、IKE が PKCS#11 ライブラリを使用して公開鍵の処理を高速化しているかどうかを示します。

```
example# ikeadm get stats
Phase 1 SA counts:
Current:  initiator:     0    responder:      0
Total:    initiator:    21   responder:      27
Attempted:initiator:    21   responder:      27
Failed:   initiator:     0   responder:       0
             initiator fails include 0 time-out(s)
PKCS#11 library linked in from /opt/system/core-osonn/lib/libpkcs11.so
example#
```

例 12　証明書キャッシュの表示

次のコマンドは証明書キャッシュを表示し、該当する場合は関連する秘密鍵のステータスも表示します。

```
example# ikeadm dump certcache
```

例 13　PKCS#11 トークンへのログイン

次のコマンドは、PKCS#11 トークンオブジェクトにログインして秘密鍵のロックを解除する方法を示したものです。

```
example# ikeadm token login "Sun Metaslot"
Enter PIN for PKCS#11 token:
ikeadm: PKCS#11 operation successful
```

終了ステータス

次の終了値が返されます。

0　正常終了。

0 以外　エラーが発生しました。対応するエラーメッセージが標準エラーに書き込まれます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	非インタフェース

関連項目

in.iked(1M), ike.config(4), ike.preshared(4), attributes(5), ipsec(7P)

Schneier, Bruce 著『Applied Cryptography: Protocols, Algorithms, and Source Code in C』、第 2 版、John Wiley & Sons, New York, NY、1996 年

注意事項

in.iked は大域ゾーンと排他的 IP ゾーンでのみ実行可能なので、このコマンドは共有 IP ゾーンでは役立ちません。

名前

ikecert – マシンのファイルシステム上の公開鍵証明書データベースを操作する

形式

```
ikecert certlocal
     [-a | -e | -h | -k | -l | -r | -U | -C | -L]
     [[-p] -T PKCS#11 token identifier]
     [option_specific_arguments]...

ikecert certdb [-a | -e | -h | -l | -r | -U | -C | -L]
     [[-p] -T PKCS#11 token identifier]
     [option_specific_arguments]...

ikecert certrldb [-a | -e | -h | -l | -r]
     [option_specific_arguments]...

ikecert tokens
```

機能説明

`ikecert` コマンドは、マシンのファイルシステム上の公開鍵証明書データベースを操作します。後述の「ファイル」の項を参照してください。

`ikecert` には、3 つの主要なリポジトリにそれぞれ対応する 3 つのサブコマンドと、使用可能なハードウェアトークンを一覧表示するサブコマンドがあります。

- `certlocal` は秘密鍵リポジトリを処理します。
- `certdb` は公開鍵リポジトリを処理します。
- `certrldb` は証明書失効リスト (CRL) リポジトリを処理します。
- `tokens` は、特定の PKCS#11 ライブラリの使用可能な PKCS#11 トークンを表示します。

サポートされている PKCS#11 ライブラリとハードウェアは、Sun 暗号化アクセラレータ 4000 だけです。

オプション

`tokens` 以外のサブコマンドにはオプションを 1 つ指定する必要があり、そのあとにオプション固有の引数をいくつか指定する場合もあります。

`tokens` サブコマンドは、`/etc/inet/ike/config` に指定されている PKCS#11 ライブラリの使用可能なトークンをすべて一覧表示します。

次のオプションがサポートされています。

`-a`

certlocal
: `certlocal` サブコマンドにこのオプションを指定すると、インターネット鍵交換 (IKE) のローカル ID データベースに秘密鍵がインストール (追加) されます。鍵データは標準入力から読み込まれ、その形式は Solaris 専用形式または暗号化されてい

ない PKCS#8 DER 形式です。鍵の形式は自動的に検出されます。PKCS#8 鍵ファイルは PEM 形式です。パスワードで保護された暗号化された形式のファイルは認識されませんが、OpenSSL で入手可能なツールを使用すると適切に変換できます。

このオプションを PKCS#11 ハードウェアオブジェクトで使用するには、対応する公開証明書が IKE データベースにすでに存在している必要があります。公開証明書と秘密鍵の両方をインポートする場合は、`certdb` サブコマンドを使用して公開部分を最初にインポートする必要があります。

certdb

`certdb` サブコマンドにこのオプションを指定すると、証明書が標準入力から読み込まれて IKE 証明書データベースに追加されます。`PEM Base64` または `ASN.1 BER` でエンコードされた `X.509` 証明書でなければなりません。証明書には、そのアイデンティティーの名前が採用されます。

このオプションでは、次のどちらかの方法で PKCS#11 ハードウェアキーストアに証明書をインポートできます。1 つは、一致する公開鍵オブジェクトと既存の秘密鍵オブジェクトの両方を `certlocal -kc` オプションで作成する方法、もう 1 つは、PKCS#11 トークンを `-T` オプションで明示的に指定する方法です。

certrldb

`certrldb` サブコマンドにこのオプションを指定すると、IKE データベースに CRL がインストール (追加) されます。CRL は標準入力から読み込まれます。

-e [-f pkcs8] *slot*

certlocal

`certlocal` サブコマンドにこのオプションを指定すると、IKE ローカル ID データベースから秘密鍵が抽出されます。鍵データは標準出力に書き込まれます。slot は、どの秘密鍵を抽出するかを指定します。秘密鍵はバイナリ BER 形式のみで抽出されます。

このオプションは特に慎重に使用してください。後述の「セキュリティー」の節を参照してください。

このオプションは PKCS#11 ハードウェアオブジェクトでは機能しません。

「`-f pkcs8`」と組み合せて使用すると、暗号化されていない PKCS#8 形式で秘密鍵が抽出されます。

-e [-f *output-format*] `certspec`

certdb

`certdb` サブコマンドにこのオプションを指定すると、certspec に一致する証明書が IKE 証明書データベースから抽出され、標準出力に書き込まれます。*output-format* オプションはエンコード形式を指定します。指定できるオプションは `PEM` と `BER` です。最初に一致し

たアイデンティティーが抽出されます。デフォルトの出力形式は PEM です。

certrldb

certrldb サブコマンドにこのオプションを指定すると、IKE データベースから CRL が抽出されます。鍵データは標準出力に書き込まれます。certspec は、どの CRL を抽出するかを指定します。データベース内で最初に一致したものが抽出されます。certspec パターンの詳細については、後述の「注意事項」を参照してください。

-kc -m *keysize* -t *keytype* -D *dname* -A *altname*[...]
[-S *validity start_time*][-F *validity end_time*]
[-T *PKCS#11 token identifier*]

certlocal

certlocal サブコマンドにこのオプションを指定すると、IKE 公開鍵/秘密鍵のペアが生成され、ローカル ID データベースに追加されます。証明書要求も生成され、標準出力に書き込まれます。上記のオプションの詳細は、*dname* 引数については「注意事項」を参照し、このコマンドの *altname* 引数については「代替名」を参照してください。

-T を指定すると、ハードウェアトークンで鍵のペアが生成されます。

-T と同時に -p を指定すると、PKCS#11 トークン PIN が平文でディスク上に保存され、ルート保護されたファイルアクセス権が設定されます。指定しない場合は、in.iked(1M) の実行開始後に ikeadm(1M) で

トークンのロックを解除する必要があります。

-ks -m *keysize* -t *keytype* -D *dname* -A *altname*[...]
[-S *validity start_time*][-F *validity end_time*]
[-f *output-format*][[-p] -T *PKCS#11 token identifier*]

certlocal

certlocal サブコマンドにこのオプションを指定すると、公開鍵/秘密鍵のペアが生成され、ローカル ID データベースに追加されます。自己署名付き証明書も生成され、証明書データベースにインストールされます。このコマンドの *dname* 引数と *altname* 引数の詳細については、後述の「注意事項」を参照してください。

-T を指定すると、ハードウェアトークンで鍵のペアが生成され、自己署名付き証明書もハードウェアに保存されます。

-l [-v] [*slot*]

certlocal

certlocal サブコマンドにこのオプションを指定すると、ローカル ID データベース内の秘密鍵が一覧表示されます。-v オプションを指定すると、出力が詳細モードに切り替わり、証明書全体が出力されます。

-v オプションは特に慎重に使用してください。後述の「セキュリティー」の節を参照してください。-v オプションは PKCS#11 ハードウェアオブジェクトでは機能しません。

-l [-v] [certspec]

certdb

certdb サブコマンドにこのオプションを指定すると、IKE 証明書データベース内の証明書が一覧表示されます。certspec を指定した場合は、そのパターンに一致するものだけが表示されます。このリストには、証明書のアイデンティティー文字列が表示され、鍵データベースに存在する場合は秘密鍵も表示されます。-v を指定すると、出力が詳細モードに切り替わり、証明書全体が出力されます。

一致する証明書がハードウェアトークン上にある場合は、トークン ID も表示されます。

certrldb

certrldb サブコマンドにこのオプションを指定すると、IKE データベース内の CRL と、データベース内にあり発行者名に一致する証明書が一覧表示されます。certspec を使用すると、特定の CRL を表示するように指定できます。-v オプションを指定すると、出力が詳細モードに切り替わり、証明書全体が出力されます。certspec パターンの詳細については、後述の「注意事項」を参照してください。

-r *slot*

certlocal

certlocal サブコマンドにこのオプションを指定すると、指定したスロットのローカル ID が削除されます。対応する公開鍵が存在する場合は削除されません。このスロットが「壊れてい

る」とみなされる場合やほかの理由で認識されない場合にも、同じく削除されます。

これを PKCS#11 ハードウェアオブジェクトに対して呼び出すと、PKCS#11 公開鍵オブジェクトと秘密鍵オブジェクトも削除されます。公開鍵オブジェクトが `certdb -r` ですでに削除されていても問題ありません。

- r certspec

certdb

IKE 証明書データベースから証明書を削除します。指定した証明書パターンに一致する証明書が削除されます。これらの証明書に対応する秘密鍵が `certlocal` データベースに存在する場合、それらは削除されません。最初に一致したアイデンティティーが削除されます。

パターンでスロットを指定すると、そのスロットが「壊れている」とみなされる場合やほかの理由で認識されない場合にも、同じく削除されます。

これを PKCS#11 ハードウェアオブジェクトに対して呼び出すと、証明書と PKCS#11 公開鍵オブジェクトも削除されます。公開鍵オブジェクトが `certlocal -r` ですでに削除されていても問題ありません。

certrldb

`certrldb` サブコマンドにこのオプションを指定すると、指定した `certspec` を持つ CRL が削除されます。

-U slot

certlocal
: certlocal サブコマンドに -T フラグとともにこのオプションを指定すると、PKCS#11 秘密鍵オブジェクトが IKE データベースからリンク解除されます。ハードウェアキーストアへのアクセスや、トークン上の秘密鍵オブジェクトの検証または削除が試みられることはありません。単にオブジェクトと IKE データベースの関連付けが解除されます。

certdb
: certdb サブコマンドに -T フラグとともにこのオプションを指定すると、PKCS#11 証明書オブジェクトが IKE データベースからリンク解除されます。ハードウェアキーストアへのアクセスや、トークン上の証明書オブジェクトや公開鍵オブジェクトの検証または削除が試みられることはありません。単にオブジェクトと IKE データベースの関連付けが解除されます。

-C certspec

certlocal
: certlocal サブコマンドにこのオプションを指定すると、秘密鍵、それに対応する証明書、および公開鍵が、ディスク上のキーストアから PKCS#11 トークンで指定されたハードウェアキーストアにコピーされます。このサブコマンドでは、一部が失敗しても、これらの各コンポーネントの作成が試みられます。いずれの場合も、ディスク上の元の秘密鍵および公開証明書はそのまま残る

ため、これらは個別に削除する必要があります。FIPS-140 準拠のデバイスなど、一部のハードウェアキーストアでは、この方法による秘密鍵オブジェクトの移行がサポートされていません。

certdb

`certdb` サブコマンドにこのオプションを指定すると、指定した `certspec` に一致する証明書、およびそれに対応する公開鍵が、ディスク上のキーストアから PKCS#11 トークンで指定されたハードウェアキーストアにコピーされます。元の公開証明書はそのまま残るため、必要な場合は個別に削除する必要があります。

`-p` を指定すると、PKCS#11 トークン PIN が平文でディスク上に保存され、ルート保護されたファイルアクセス権が設定されます。指定しない場合は、`in.iked(1M)` の実行開始後に `ikeadm(1M)` でトークンのロックを解除する必要があります。

-L pattern

certlocal

`certlocal` サブコマンドにこのオプションを指定すると、トークン上に存在している秘密鍵オブジェクトが IKE データベースにリンクされます。オブジェクト自体はトークン上に残ります。このオプションは、オブジェクトが元から Solaris の IKE ユーティリティーでトークン上に作成されたかのように、オブジェクトの

存在を IKE インフラストラクチャーに認識させるだけです。

certdb

`certdb` サブコマンドにこのオプションを指定すると、トークン上に存在している証明書オブジェクトが IKE データベースにリンクされます。オブジェクト自体はトークン上に残ります。このオプションは、オブジェクトが元から Solaris の IKE ユーティリティーでトークン上に作成されたかのように、オブジェクトの存在を IKE インフラストラクチャーに認識させるだけです。

`-p` を指定すると、PKCS#11 トークン PIN が平文でディスク上に保存され、ルート保護されたファイルアクセス権が設定されます。省略した場合は、`in.iked(1M)` の実行開始後に `ikeadm(1M)` でトークンのロックを解除する必要があります。

パラメータ

サポートされているパラメータは次のとおりです。

certspec

証明書仕様のパターンマッチングを指定します。`certspec` には、被認証者名、発行者名、および被認証者代替名を指定できます。

これらを使用すると、指定の `certspec` 値に一致し、ほかの `certspec` 値には一致しない証明書を指定できます。証明書に存在しないと想定する `certspec` 値を表すには、タグの前に ! を付加します。

有効な `certspec` は次のとおりです。

```
<Subject Names>
SUBJECT=<Subject Names>
ISSUER=<Issuer Names>
SLOT=<Slot Number in the certificate database>

Example:"ISSUER=C=US, O=SUN" IP=1.2.3.4 !DNS=example.com
Example:"C=US,   O=CALIFORNIA" IP=5.4.2.1 DNS=example.com
```

代替名の有効な引数は次のとおりです。

```
IP=<IPv4 address>
DNS=<Domain Name Server address>
EMAIL=<email (RFC 822) address>
URI=<Uniform Resource Indicator value>
DN=<LDAP Directory Name value>
RID=<Registered Identifier value>
```

キーワードタグなしで、有効なスロット番号を指定できます。代替名はキーワードタグを使用して発行することもできます。

-A
証明書の被認証者代替名。-A オプションに続く引数は、*tag=value* という形式で指定するようにしてください。有効なタグは、IP、DNS、EMAIL、 URI、DN、および RID です (下記の例を参照)。

-D
証明書の被認証者の X.509 識別名。一般的な形式は次のとおりです。C=country (国)、O=organization (組織)、OU=organizational unit (組織単位)、CN=common name (共通名)。有効なタグは、C、 O、OU、および CN です。

-f
エンコード出力形式。pem (PEM Base64) または ber (ASN.1 BER) です。 -f を指定しない場合、 pem とみなされます。

-F *validity end_time*
証明書の有効期間の終了時刻。-F フラグを指定しない場合、 有効期間の終了時刻は有効期間の開始時刻から 4 年として計算されます。有効期間の日付と時刻の構文については、「注意事項」を参照してください。

-m
鍵のサイズ。有効な値は、512、1024、2048、3072、または 4096 です。Solaris 暗号化フレームワークでサポートされている鍵サイズを調べるには、次のコマンドを使用します。

```
% cryptoadm list -vm
```

このコマンドで表示されるメカニズムについては、pkcs11_softtoken(5) を参照してください。ハードウェアアクセラレーションを備えたシステムの場合、ハードウェアでサポートされているメカニズムがプロバイダごとに個別のセクションに表示されます。メカニズムは次のいずれかにできます。

```
CKM_RSA_PKCS_KEY_PAIR_GEN
CKM_DSA_KEY_PAIR_GEN
CKM_DH_PKCS_KEY_PAIR_GEN
```

注 – すべてのハードウェアですべての鍵サイズがサポートされているわけではありません。たとえば、Sun 暗号化アクセラレータ 4000 のキーストアでは (後述の -T オプションを使用した場合)、RSA には最大 2048 ビット、DSA には最大 1024 ビットの鍵だけがサポートされています。

-S *validity start_time*
証明書の有効期間の開始時刻。-S フラグを指定しない場合、 有効期間の開始時刻として現在の日付と時刻が使用されます。有効期間の日付と時刻の構文については、後述の「注意事項」を参照してください。

-t
鍵のタイプ。有効な値は、rsa-sha1、rsa-md5、dsa-sha1 のいずれかです。

-T
ハードウェアキーストアの PKCS#11 トークン識別子。これは、PKCS#11 標準に準拠したハードウェアデバイスインスタンスを指定します。/etc/inet/ike/config に PKCS#11 ライブラリが指定されている必要があります。(ike.config(4) を参照。)

トークン識別子は、スペースで詰められた 32 文字の文字列です。指定したトークンの長さが 32 文字未満の場合は、自動的にスペースで詰められます。

システムに複数の PKCS#11 ライブラリが存在する場合、/etc/inet/ike/config には一度に 1 つのライブラリしか指定できないことに留意してください。1 つの PKCS#11 ライブラリインスタンスに複数のトークン (それぞれが別個のキーストアを持つ) が存在することもあります。

セキュリティー

このコマンドは、公開鍵と秘密鍵のペアから秘密鍵をファイルに保存できます。秘密鍵が外部にさらされると、悪意のある者が何らかの方法でこの鍵を入手した場合、危殆化につながる可能性があります。

PKCS#11 ハードウェアオブジェクトの機能は、ディスク上の秘密鍵の短所に一部対処できます。IKE はシステムサービスなので、ブート時のユーザーの介入は望ましくありません。ただし、トークンの PIN は必要です。したがって、PKCS#11 トークンの PIN は、ディスク上の暗号化鍵が通常置かれる場所に保存されます。このような設計上の決定は妥当と考えられます。なぜなら、ハードウェアキーストアでは、鍵の「所有」はまだ不可能で、ホストが危殆化した場合に鍵の「使用」だけが問題になるからです。PIN を越えると、ikecert のセキュリティーは PKCS#11 実装のセキュリティーまで下がります。PKCS#11 の実装も詳細に検討されるべきです。

詳細については、Bruce Schneier 著『Applied Cryptography: Protocols, Algorithms, and Source Code in C』の Matt Blaze による後書きを参照してください。

使用例

例 1 自己署名証明書の生成

次に、自己署名付き証明書の例を示します。

```
example# ikecert certlocal -ks -m 512 -t rsa-md5 -D "C=US, O=SUN" -A
IP=1.2.3.4
Generating, please wait...
Certificate generated.
Certificate added to database.
-----BEGIN X509 CERTIFICATE-----
```

例 1　自己署名証明書の生成　　*(続き)*


```
MIIBRDCB76ADAgECAgEBMA0GCSqGSIb3DQEBBAUAMBsxCzAJBgNVBAYTAlVTMQww
CgYDVQQKEwNTVU4wHhcNMDEwMzE0MDEzMDM1WhcNMDUwMzE0MDEzMDM1WjAbMQsw
CQYDVQQGEwJVUzEMMAoGA1UEChMDU1VOMFowDQYJKoZIhvcNAQEBBQADSQAwRgJB
APDhqpKgjgRoRUr6twTMTtSuNsReEnFoReVer!ztpXpQK6ybYlRH18JIqU/uCV/r
26R/cVXTy5qc5NbMwA40KzcCASOjIDAeMAsGA1UdDwQEAwIFoDAPBgNVHREECDAG
hwQBAgMEMA0GCSqGSIb3DQEBBAUAA0EApTRD23KzN95GMvPD71hwwClukslKLVg8
f1xm9ZsHLPJLRxHFwsqqjAad4j4wwwriiUmGAHLTGB0lJMl8xsgxag==
-----END X509 CERTIFICATE-----
```


例 2　CA 要求の生成

CA 要求の生成は自己署名付き証明書の生成と同じに見えます。これら 2 つの相違点は、オプション -s の代わりに -c を使用し、証明書データが CA 要求であるということだけです。

```
example# ikecert certlocal -kc -m 512 -t rsa-md5 \
   -D "C=US, O=SUN" -A IP=1.2.3.4
```

例 3　ハードウェアキーストアを使用した CA 要求

次の例は、-T オプションでトークンを指定する方法を示しています。

```
example# # ikecert certlocal -kc -m 1024 -t rsa-md5 -T vca0-keystore \
  -D "C=US, O=SUN" -A IP=1.2.3.4
```

終了ステータス

次の終了値が返されます。

0
　正常終了。

0 以外
　エラーが発生しました。対応するエラーメッセージが標準エラーに書き込まれます。

ファイル

`/etc/inet/secret/ike.privatekeys/*`
　秘密鍵。秘密鍵に対応する同じファイル名の公開鍵証明書が `/etc/inet/ike/publickeys/` に存在する必要があります。

`/etc/inet/ike/publickeys/*`
　公開鍵証明書。名前については、対応する秘密鍵の名前との関連だけが重要です。

`/etc/inet/ike/crls/*`
　公開鍵証明書失効リスト。

`/etc/inet/ike/config`
　PKCS#11 ライブラリのパス名を調べるために参照されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

ikeadm(1M), in.iked(1M), getdate(3C), ike.config(4), attributes(5), pkcs11_softtoken(5)

Schneier, Bruce 著『Applied Cryptography: Protocols, Algorithms, and Source Code in C』、第 2 版、John Wiley & Sons 発行、New York, NY、1996.

『PKCS#11 v2.11: Cryptographic Token Interface Standards』、RSA Labs 発行、2001 年 11 月

注意事項

次に、-F または -S フラグを使用する場合の有効期間の日付と時刻の構文について説明します。

相対日付の場合、構文は次のとおりです。

```
{+,-}[Ns][Nm][Nh][Nd][Nw][NM][Ny]
```

各表記の意味は次のとおりです。

N
: 整数を表します

s
: 秒を表します

m
: 分を表します

h
: 時間を表します

d
: 日を表します

w
: 週を表します

M
: 月を表します

y
: 年を表します

これらのパラメータは任意の順序で指定できます。たとえば、「+3d12h」は今から 3.5 日後、「-3y2M」は 3 年 2 か月前です。

固定値を持つパラメータはすべて、秒数の絶対値で足し合わせることができます。月と年は、秒数が可変なので、カレンダを使用して計算されます。月と年は、長さが固定ではないため、1 年または 1 か月を足すことは、次の年または次の月の同じ日を表すものとして定義されます。たとえば、今日が 2005 年 1 月 26 日で、今日から 3 年 1 か月後に証明書が失効する場合、失効日 (有効期間の終了時刻) は 2008 年 2 月 26 日になります。オーバーフローは適切に処理されます。たとえば、2 月には 28 日しかないため、2005 年 1 月 31 日から 1 か月後は 2005 年 3 月 3 日です。

絶対日付の場合は、ファイル /etc/datemsk に含まれている日付形式の構文を使用できます (詳細は、getdate(3C) を参照)。日付文字列の前に「+」または「-」が付加されている場合は、現在の時刻からの相対時刻とみなされます。それ以外の場合は、絶対日付とみなされます。有効期間の終了日時が有効期間の開始日時より大きいことを確認する妥当性検査も行われます。たとえば、次のコマンドを実行すると、1 日と 2 時間前を開始日時、2007 年 1 月 22 日のローカル時間 12:00:00 を終了日時として証明書が作成されます。

```
# ikecert certlocal -ks -t rsa-sha1 -m 1024 \
    -D "CN=mycert, O=Sun, C=US" \
    -S -1d2h -F "01/22/2007 12:00:00"
```

in.iked(1M) は大域ゾーンと排他的 IP ゾーンでのみ実行可能なので、このコマンドは共有 IP ゾーンでは役立ちません。

名前

ilbadm – 負荷分散規則の確立と操作

形式

```
ilbadm create-rule [-e] [-p] -i vip=value,port=value[,protocol=value]
     -m lbalg=value,type=value[,proxy-src=ip-range][,pmask=mask]
     [-h hc-name=value[,hc-port=value]]
     [-t [conn-drain=N][,nat-timeout=N],[persist-timeout=N]]
     -o servergroup=value name

ilbadm show-rule [-e|-d] [-f |[-p] -o key[,key ...]] [name ...]

ilbadm delete-rule -a | name ...

ilbadm enable-rule [name ...]

ilbadm disable-rule [name ...]

ilbadm show-statistics [-p] -o field[,field] [-thAdvi]
     [-r rulename] | [-s servername] [interval [count]]

ilbadm create-servergroup [-s server=hostspec[:portspec...]] groupname

ilbadm delete-servergroup groupname

ilbadm show-servergroup [-s|-f|[-p] -o field[,field]] [[-v] name]

ilbadm enable-server server ...

ilbadm disable-server server ...

ilbadm show-server [[-p] -o field[,field...]] [rulename...]

ilbadm add-server -s server=value[,value ... ] name

ilbadm remove-server -s server=value[,value ... ] name

ilbadm create-healthcheck [-n] -h hc-test=value
     [,hc-timeout=value][,hc-count=value][,hc-interval=value] hcname

ilbadm delete-healthcheck hcname

ilbadm show-healthcheck [hcname ...]

ilbadm show-hc-result [rule-name]

ilbadm show-nat [count]

ilbadm show-persist [count]

ilbadm export-config filename

ilbadm import-config [-p] filename
```

機能説明

`ilbadm` コマンドは、後述のサブコマンドを使用して、統合ロードバランサ (ILB) の規則の操作および情報の表示を行います。

規則名の大文字と小文字は区別されませんが、入力したとおりに保持されます。規則名の長さは 19 文字に制限されます。サーバー名は 14 文字を超えることはできません。

-p オプションで呼び出す機械可読な出力では、どの場合も、出力または表示するフィールドを -o オプションで指定する必要があります。フィールドは、コマンド行に現れたとおりの順序で表示されます。複数のフィールドを指定する場合は、コロン (:) 文字で区切ります。表示された文字列自体にコロンまたはバックスラッシュ (\) が含まれる場合は、その前にバックスラッシュが付きます。機械可読な出力の場合、ヘッダーは表示されません。

サーバー ID は、create-servergroup または add-server サブコマンドでサーバーが追加されるときに、システムによって生成されます。

サーバー ID はサーバーグループ内で一意であることが保証されます。規則は 1 つのサーバーグループのみに結びつけることができるため、サーバー ID は規則ごとにも一意になります。同じサーバーグループに複数の規則を結びつけることができるため、サーバー ID 単独では規則を識別するのに不十分です。

サーバー ID とホスト名を区別できるように、サーバー ID の前には下線 (_) が付加されます。

後述のとおり、サーバーグループおよび健全性検査のエンティティーは、create-rule サブコマンドで使用する前に定義しておく必要があります。

サブコマンド

ilbadm のサブコマンドおよび関連するオプションとオペランドは次のとおりです。サブコマンドには create-rule と create-rl のように通常形式と短形式があり、数文字分の入力を省略できます。

create-rule|create-rl [-e] [-p] -i *incoming* -m *method_attributes* -o *outgoing_spec* [- h *healthcheck*] [-t *timers*] *name*

指定された一連の特性を使用して、規則 *name* を作成します。*incoming* と *method_attributes* は、一連の *key=value* ペアで指定します。 *name* がすでに存在している場合、このコマンドは失敗します。ある特定の組 (仮想 IP アドレス、ポート、およびプロトコル) が別の規則に一致する場合も、このコマンドは失敗します。create-rule の次のオプションは、コマンドの全体的な効果を制御します。

-e　create-rule 機能を有効にします。デフォルトでは、create-rule は無効になっています。

-p　規則を永続的 (スティッキー) なものとして作成します。デフォルトでは、規則が存続するのは現在のセッションの間のみです。

キーと値は 1 文字の識別子によって指定します。このような識別子および関連するキーと設定可能な値は次のとおりです。

-i

受信パケットの一致条件を指定します。

vip

(仮想) 着信先 IP アドレス

port[-port]
ポートの番号または名前。telnet、dns など。ポートは、ポート番号または記号名 (/etc/services の記述どおり) で指定できます。ポート番号の範囲もサポートされています。

protocol
TCP (デフォルト) または UDP (/etc/services を参照)。

-m
パケットの処理方法を記述するキーを指定します。

lbalg
デフォルトは roundrobin (短形式: rr) です。ほかに指定できる値は、hash-ip (短形式: hip)、hash-ip-port (短形式: hipp)、hash-ip-vip (短形式: hipv) です。

type
ネットワークのトポロジを指します。指定できる値は、DSR (dsr または d も可)、NAT (n または nat も可)、HALF-NAT (h または half-nat も可) です。

proxy-src
フル NAT の場合のみ必要です。プロキシの発信元アドレスの範囲として使用する IP アドレスの範囲を指定します。範囲は 10 個の IP アドレスに制限されます。

pmask
省略可能。stickiness という別名があります。この規則を永続的なものとして指定します。引数は CIDR 表記の接頭辞の長さです。つまり、IPv4 の場合は 0–32、IPv6 の場合は 0–128 です。このキーワードを指定するには、-p オプションを使用します。

-o
-i 節で指定された条件に一致するパケットの着信先を指定します。この識別子には 1 つの既知の引数があります。

servergroup 単一のサーバーグループをターゲットとして指定します。サーバーグループはすでに作成されている必要があります。

-h
健全性検査のオプションには 2 つの引数があります。

hc-name
定義済みの健全性検査方法の名前を指定します。

hc-port
HC テストプログラムで検査するポートを指定します。値には、キーワード ALL または ANY、あるいはサーバーグループのポート範囲内にある特定のポート番号を指定できます。

-t
カスタマイズされたタイマーを秒単位で指定します。値 0 は、システムのデフォルト値を使用することを意味します。-t に使用できる修飾子は次のとおりです。

conn-drain
サーバーの type が NAT または HALF-TYPE の場合、規則からのサーバーの削除に続いて、conn-drain がタイムアウトすると、サーバーの接続状態が削除されます。この削除は、サーバーがアイドル状態でない場合でも実行されます。

TCP のデフォルトでは、接続が正常にシャットダウンされるまで、接続状態は安定のままです。UDP のデフォルトでは、接続のアイドル状態が nat-timeout の期間に達するまでは、接続状態は安定のままです。

nat-timeout
NAT およびハーフ NAT タイプの接続だけに適用されます。接続のアイドル状態が nat-timeout の期間に達すると、接続状態が削除されます。デフォルトは、TCP の場合は 120、UDP の場合は 60 です。

persist-timeout
永続的マッピングが有効になっている場合、数値のみのマッピングは、使用されていない状態が persist-timeout 秒間続くと削除されます。デフォルトは 60 です。

サーバーグループおよび健全性検査は、create-rule で使用する前に定義しておく必要があります。

delete-rule|delete-rl -a *name*[...]
規則 *name* に関連するすべての情報を削除します。*name* が存在しない場合、このコマンドは失敗します。delete-rule には 1 つのオプションがあります。

-a
すべての規則を削除します。(*name* は無視されます。)

enable-rule|enable-rl *name*[...]
指定された規則を有効にします。名前が指定されていない場合はすべての規則を有効にします。 すでに有効になっている規則には影響はありません。

disable-rule|disable-rl *name*[...]
規則名が指定されている場合はその規則を、指定されていない場合はすべての規則を無効にします。すでに無効になっている規則を有効にしても、影響はありません。

show-statistics|show-stats [[-p] -o *field*[,...]] [-tv] [-A | -d] [[-i] -r *rulename* | - s *servername*] [*interval* [*count*]]
統計を表示します。統計の出力は、次に示すオプションで制御されます。このサブコマンドの構文とセマンティクスは、vmstat(1M) に基づいています。

`-t`
すべてのサンプルの前にタイムスタンプを付加します。

`-d`
全期間にわたる差分を表示します。デフォルトは、1 秒あたりの変化です。`-a` オプションと同時には使用できません。

`-A`
絶対的な数値を表示します。つまり、モジュールの初期化、規則の作成、およびサーバーの追加以降の数値です。`-d` オプションと同時には使用できません。

`-r` *rulename*
指定された *rulename* のみに関する統計を表示します。`-i` オプションと組み合わせると、サーバーごとに 1 行が表示されます。

`-s` *servername*
server のみに関する統計を表示します。`-i` オプションと組み合わせると、規則ごとに 1 行が表示されます。

`-i`
`-r` オプションと `-s` オプションで表示される情報を項目化します。`-i` はこれらのオプションとの組み合わせでのみ有効です。`-v` オプションと同時には機能しません。

`-v`
ドロップに関するより詳細な情報を表示します。規則名が指定されている場合、ドロップはサーバーごとではなく規則ごとにカウントされます。`-i` オプションと同時には機能しません。

`-p`
機械可読な形式を表示します。`-o` オプションを使用する必要があります。

`-o` *field*
次のリストから 1 つまたは複数を指定できます。*field* は大文字でも小文字でもかまいません。

`PKT_P`	処理されたパケット。
`BYTES_P`	処理されたバイト。
`PKT_U`	未処理のパケット。
`BYTES_U`	未処理のバイト。
`PKT_D`	ドロップされたパケット。
`BYTES_D`	ドロップされたバイト。
`ICMP_P`	処理された ICMP エコー要求。
`ICMP_D`	ドロップされた ICMP エコー要求。

ICMP2BIG_P　ICMP 断片化が必要 (処理されたメッセージ)。

ICMP2BIG_D　断片化が必要 (ドロップされたメッセージ)。

NOMEMP_D　メモリー不足状態のためドロップされたパケット。

NOPORTP_D　使用可能な発信元ポートがなかったため NAT モードでドロップされたパケット。

疑問符 (?) が列のエントリとして表示される場合は、正しい値を判定できないことを示しています。その多くは、規則またはサーバーが追加または削除されたことが原因で発生します。

ヘッダーは 10 サンプルごとに 1 回表示されます。タイムスタンプの形式は、C ロケールの date(1) 形式に従います。規則の追加や削除は検出されません。

show-rule|show-rl [-d|-e] [-f| [-p] -o *field*[,...]] [*name*...]

規則が指定されている場合はそれらの規則の特性を、指定されていない場合はすべての規則の特性を表示します。このサブコマンドには、次のオプションがあります。

-d
無効になっている規則だけを表示します。

-e
有効になっている規則だけを表示します。

-f
完全なリストを表示します。

-o *field*[,...]
field の出力を表示します。-f オプションと同時には使用できません。

-p
「機能説明」で説明されている形式で、解析可能な出力を表示します。-o オプションが必要です。

(-p の有無にかかわらず) -o オプションと -f オプションを同時に指定することはできません。

show-nat *count*

NAT テーブルの情報を表示します。*count* が指定されている場合は、NAT テーブルから *count* 件のエントリを表示します。count が指定されていない場合は、NAT テーブル全体を表示します。

count

このコマンドを連続して実行した場合の要素の相対位置については想定を設けないでください。たとえば、show-nat 10 を 2 回実行したときに、2 回とも同じ 10 項目が表示される保証はありません。処理量の多いシステムでは特にそうです。

表示形式:

```
T: IP1 > IP2 >>> IP3 > IP4
```

これらの項目について、次に説明します。

T　このエントリで使用されているトランスポートプロトコル。

IP1　クライアントの IP アドレスとポート。

IP2　VIP とポート。

IP3　ハーフ NAT モードの場合は、クライアントの IP アドレスとポート。フル NAT モードの場合は、NAT 処理されたクライアントの IP アドレスとポート。

IP4　バックエンドサーバーの IP アドレスとポート。

show-persist|show-pt *count*

持続性テーブルの情報を表示します。*count* が指定されている場合は、テーブルから *count* 件のエントリを表示します。count が指定されていない場合は、持続性テーブル全体を表示します。

このコマンドを連続して実行した場合の要素の相対位置については想定を設けないでください。たとえば、show-persist 10 を 2 回実行したときに、2 回とも同じ 10 項目が表示される保証はありません。処理量の多いシステムでは特にそうです。

表示形式:

```
R: IP1 --> IP2
```

これらの項目について、次に説明します。

R

この持続性エントリが関連付けられている規則。

IP1

クライアントの IP アドレスとポート。

IP2

バックエンドサーバーの IP アドレス。

export-config|export-cf [*filename*]

現在の構成を ilbadm import による再インポートに適した形式でエクスポートします。ファイル名が指定されていない場合、このサブコマンドは標準出力に書き出します。

import-config|import-cf [-p] [*filename*]

ファイルから構成の内容を読み込みます。デフォルトでは、この操作によって既存の構成は上書きされます。ファイル名が指定されていない場合、このサブコマンドは標準入力から読み込みます。このサブコマンドには、次のオプションがあります。

`-p`
既存の構成を維持し、増分インポートを実行します。

`create-servergroup`|`create-sg` [`-s` *server=hostspec*[:*portspec*...]] *groupname*
サーバーグループを作成します。あとで `add-server` サブコマンドを使用するとサーバーを追加できます。規則の作成時にバックエンドサーバーを指定するために使用できるエンティティーはサーバーグループだけです。指定されたサーバーグループが 1 つ以上の規則に関連付けられている場合、サーバーは追加されたときに有効になります。このサブコマンドには、次のオプションとオペランドがあります。

`-s` *server=hostspec*[:*portspec*...]
サーバーグループに追加するサーバーのリストを指定します。

hostspec はホスト名または IP アドレスです。IPv6 アドレスは、「:*portspec*」と区別するために角括弧 (`[]`) で囲む必要があります。

portspec はサービス名またはポート番号です。ポート番号が指定されていない場合は、1–65535 の範囲内の数が使用されます。

`disable-server`|`disable-srv` *server*
1 つまたは複数のサーバーを無効にします。つまり、このサーバーにトラフィックを転送しないようカーネルに指示します。`disable-server` は、このサーバーが属しているサーバーグループに割り当てられているすべての規則に適用されます。

server はサーバー ID です。

`enable-server`|`enable-srv` *server*...
無効になっているサーバーを再度有効にします。

`show-server`|`show-srv` [[`-p`] `-o` *field*[,*field*...]] [*rulename*...]
指定された規則に関連付けられているサーバーを表示します。規則名が指定されていない場合はすべてのサーバーを表示します。このサブコマンドには、次のオプションがあります。

`-o` *field*[,*field*...]
指定されたフィールドだけを表示します。

`-p`
フィールドを解析可能な形式で表示します。`-o` オプションが必要です。

`delete-servergroup`|`delete-sg` *groupname*
サーバーグループを削除します。

`show-servergroup`|`show-sg` [[`-p`] `-o` *field*[,...]] [*name*]
name が指定されている場合はそのサーバーグループを、指定されていない場合はすべてのサーバーグループを表示します。このサブコマンドには、次のオプションがあります。

`-o` *field*[,...]
field の出力を表示します。

`-p`
「機能説明」で説明されている形式で、解析可能な出力を表示します。`-o` オプションが必要です。

`add-server`|`add-srv` `-s` *server*=*value*[, *value*...] *servergroup*
指定されたサーバーを *servergroup* に追加します。*value* の定義については、`create-servergroup` の説明を参照してください。

`-s`
`create-servergroup` を参照してください。

サーバーグループに対して `remove-server` を実行した直後は、接続のドレインが完了していないため、そのサーバーグループに対して `add-server` を実行すると失敗することがあります。この失敗を防ぐ方法については、`remove-server` サブコマンドの説明を参照してください。

`remove-server`|`remove-srv` `-s` *server*=*value*[, *value*...] *servergroup*
指定されたサーバーを *servergroup* から削除します。

`-s`
1 つまたは複数のサーバー ID。

サーバーが NAT またはハーフ NAT 規則で使用されている場合は、サーバーを削除する前に、`disable-server` を使用してサーバーを無効にすることをお勧めします。サーバーを無効にすると、サーバーは接続ドレイン状態になります。すべての接続のドレインが完了したあと、`remove-server` でサーバーを削除できます。`conn-drain` タイムアウト値が設定されている場合は、このタイムアウト期間が完了すると接続ドレイン状態は終了します。デフォルトの `conn-drain` タイムアウト値は 0 で、接続が正常にシャットダウンされるまで待つことを意味します。

`create-healthcheck`|`create-hc` [`-n`] `-h` hc-test=*value*,hc-timeout=*value*, hc-count=*num_value*,hc-interval=*value hcname*
規則で使用する健全性検査オブジェクトを設定します。規則に関連付けられているすべてのサーバーが、同じテストを使用して検査されます。サーバーの健全性検査イベントでは、1 回から `hc-count` 回まで `hc-test` が実行されます。ある `hc-test` の結果、サーバーが応答していない場合は、さらに `hc-test` 検査が実行され、呼び出しが `hc-count` 回に達するとサーバーはダウンしているとみなされます。

`-h`
`hc-test` は、成功するか `hc-timeout` のタイムアウトまで、最大 `hc-count` 回実行されます。ある特定の規則で、そのすべてのサーバーが同じテストを使用して検査されます。テストは次のとおりです。

hc-test
: PING、TCP、外部メソッド (スクリプトまたはバイナリ)。外部メソッドはフルパス名で指定するようにしてください。

hc-timeout
: hc-test の一時的な失敗のあとにこのしきい値に達すると、テストは失敗とみなされます。hc-test テストを強制終了すると、結果は失敗とみなされます。デフォルト値は 5 秒です。

hc-count
: hc-test の実行を試みる最大回数。この回数に達すると、サーバーはダウンしているとみなされます。デフォルト値は 3 回です。

hc-interval
: hc-test の呼び出し間隔。この値は、hc-timeout x hc-count よりも大きくする必要があります。デフォルト値は 30 秒です。

外部メソッドには次の引数が渡されます。

$1
: VIP (IPv4 または IPv6 リテラルアドレス)。

$2
: サーバー IP (IPv4 または IPv6 リテラルアドレス)。

$3
: プロトコル (文字列で UDP、TCP)。

$4
: 負荷分散モード、DSR、NAT、HALF_NAT。

$5
: 数値のポート。

$6
: メソッドが失敗を返すまでに待機する最大時間 (秒単位)。これを超えて実行し続けると、メソッドは強制終了されることがあります。その場合、テストは失敗とみなされます。

外部メソッドでは、成功の場合は 0 (またはバックエンドサーバーまでの往復時間をマイクロ秒で表した値) を返し、サーバーがダウンしているとみなされる場合は -1 を返すようにしてください。

より上のレイヤーの健全性検査、TCP、UDP、および外部テストを開始する前に、デフォルトの ping テストが実行されます。ping が失敗した場合は、上のレイヤーのテストは実行されません。これらの上のレイヤーの健全性検査でデフォルトの ping 検査を無効にするには、-n を使用します。

-n
: 上のレイヤーの健全性検査でデフォルトの ping テストを無効にします。

`delete-healthcheck`|`delete-hc` *hcname*...
指定された健全性検査オブジェクト (*hcname*) を削除します。指定された健全性検査オブジェクトが有効になっている規則に関連付けられている場合、オブジェクトの削除は失敗します。

`show-healthcheck`|`show-hc` [*hcname*...]
指定された健全性検査 (*hcname*) の健全性検査情報を一覧表示します。健全性検査が指定されていない場合は、既存のすべての健全性検査に関する情報が表示されます。

`show-hc-result`|`show-hc-res` [*rule-name*]
rule-name に関連付けられているサーバーの健全性検査結果を一覧表示します。*rule-name* が指定されていない場合は、すべてのサーバーの健全性検査結果が表示されます。

使用例

例 1　NAT モードの構成

次のコマンドは、健全性検査とタイマーを設定して規則を作成します (ポート範囲のシフトとセッションの持続性)。

```
# ilbadm create-healthcheck -h hc-test=tcp,hc-timeout=2,hc-count=3, \
     hc-interval=10 hc1
# ilbadm create-servergroup -s \
     server=60.0.0.10:6000-6009,60.0.0.11:7000-7009 sg1
# ilbadm create-rule -e -i vip=81.0.0.10,port=5000-5009,protocol=tcp \
-m lbalg=rr,type=NAT,proxy-src=60.0.0.101-60.0.0.104, \
     pmask=24 \
-h hc-name=hc1 \
-t conn-drain=180,nat-timeout=180,persist-timeout=180 \
-o servergroup=sg1 rule1
```

次のコマンドは、健全性検査は設定せずに、デフォルトのタイマー値で規則を作成します。

```
# ilbadm create-servergroup -s server=60.0.0.10 sg1
 # lbadm create-rule -e -i vip=81.0.0.10,port=5000 \
     -m lbalg=rr,type=NAT,proxy-src=60.0.0.105 \
     -o servergroup=sg1 rule1
# ilbadm add-server -e -s server=60.0.0.11sg1
# ilbadm enable-rule rule1
```

例 2　ハーフ NAT モードの構成

次のコマンドは、ハーフ NAT モードを構成し、ポート範囲を短縮します。

```
# ilbadm create-servergroup sg1
# ilbadm create-rule -e -i vip=81.0.0.10,port=5000-5009 \
     -m lbalg=rr,type=h -o servergroup=sg1 rule1
# ilbadm add-server -s server=60.0.0.10:6000,60.0.0.11:7000 sg1
```

例 3　DSR モードの構成と 2 つの規則セットの準備

次のコマンドは、HTTP と FTP トラフィックの間の負荷分散を有効にする 2 つの規則セットを確立します。どちらのタイプのトラフィックもインタフェース 60.0.0.10 を通ります。

```
# ilbadm create-servergroup -s servers=60.0.0.9,60.0.0.10 websg
# ilbadm create-servergroup -s servers=60.0.0.10,60.0.0.11 ftpgroup

# ilbadm create-rule -e -i vip=81.0.0.10,port=80 \
     -m lbalg=hash-ip-port,type=DSR \
     -o servergroup=websg webrule
# ilbadm create-rule -e -i vip=81.0.0.10,port=ftp \
     -m lbalg=hash-ip-port,type=DSR,pmask=24 \
     -o servergroup=ftpgroup ftprule
# ilbadm create-rule -e -p -i vip=81.0.0.10,port=ftp-data \
     -m lbalg=hash-ip-port,type=DSR,pmask=24 \
     -o servergroup=ftpgroup ftpdatarule
```

例 4　規則、サーバーグループ、および健全性検査の削除

次のコマンドは、最初の例で確立された規則、サーバーグループ、および健全性検査を削除します。

```
# ilbadm ilbadm delete-rule -a
# ilbadm delete-servergroup sg1
# ilbadm delete-healthcheck hc1
```

例 5　規則の一覧表示

次のコマンドは、規則の一覧を表示します。

```
# ilbadm show-rule
RULENAME            STATUS LBALG      TYPE    PROTOCOL VIP  PORT
r2                  E     hash-ip     NAT     TCP 45.0.0.10 81
r1                  E     hash-ip     NAT     TCP 45.0.0.10 80

# ilbadm show-rule -f
       RULENAME: rule1
         STATUS: E
           PORT: 80
       PROTOCOL: TCP
          LBALG: roundrobin
           TYPE: HALF-NAT
      PROXY-SRC: --
        PERSIST: --
        HC-NAME: hc1
        HC-PORT: ANY
     CONN-DRAIN: 0
    NAT-TIMEOUT: 120
PERSIST-TIMEOUT: 60
```

例 5 規則の一覧表示 (続き)

```
    SERVERGROUP: sg1
            VIP: 80.0.0.2
        SERVERS: _sg1.0,_sg1.1
```

例 6 規則のエクスポートとインポート

次の各コマンドは、標準出力を対象にした規則のエクスポートとインポート、およびファイルを対象にした規則のエクスポートとインポートの方法を示しています。

```
# ilbadm export-config

create-servergroup ftpgroup
add-server -s server=10.1.1.3:21 ftpgroup
add-server -s server=10.1.1.4:21 ftpgroup
create-servergroup webgroup_v6
add-server -s server=[2000::ff]:80 webgroup_v6
create-rule -e protocol=tcp,VIP=1.2.3.4,port=ftp \
        -m lbalg=roundrobin,type=DSR \
        -o servergroup=ftpgroup rule4
create-rule protocol=tcp,VIP=2003::1,port=ftp \
        -m lbalg=roundrobin,type=DSR \
        -o servergroup=ftpgroup6 rule3
create-rule -e protocol=tcp,VIP=2002::1,port=http \
        -m lbalg=roundrobin,type=DSR \
        -o serverrgroup=webgrp_v6 RULE-all
```

次のコマンドは、規則をファイルにエクスポートします。

```
# ilbadm export-config /tmp/ilbrules
```

このコマンドのあと、/tmp/ilbrules には、1 つ前のコマンドで表示される出力が保存されています。

次のコマンドは、規則をファイルからインポートします。

```
# ilbadm import-config /tmp/ilbrules
```

このコマンドを実行すると、配置されているすべての規則が /tmp/ilbrules の内容で置き換えられます。

次のコマンドは、規則を標準入力からインポートします。

```
# cat /tmp/ilbrules | ilbadm import-config
```

このコマンドの効果は、1 つ前のコマンドの効果と同じです。

例 7 単一の健全性検査の作成

次のコマンドは、単一の健全性検査を作成します。

例 7　単一の健全性検査の作成　　*(続き)*

```
# ilbadm create-healthcheck -h hc-timeout=3,hc-count=2,hc-interval=8,\
     hc-test=tcp hc1
```

例 8　すべての健全性検査の一覧表示

次のコマンドは、既存のすべての健全性検査を一覧表示します。

```
# ilbadm show-healthcheck
HCNAME        TIMEOUT COUNT   INTERVAL DEF_PING TEST
hc1           2       1       10       Y        tcp
hc2           2       1       10       N        /usr/local/bin/probe
```

例 9　単一の健全性検査の削除

次のコマンドは、単一の健全性検査を削除します。

```
# ilbadm delete-healthcheck hc1
```

例 10　統計情報の表示

次のコマンドは、統計情報を 1 秒間隔で合計 3 回表示します。

```
# ilbadm show-stats -A 1 3
 PKT_P   BYTES_P   PKT_U   BYTES_U   PKT_D   BYTES_D
0       0         0       0         4       196
0       0         0       0         4       196
0       0         0       0         4       196
```

統計情報を詳細モードで 1 秒間隔で表示する場合は、次のようなコマンドを使用します。出力の幅が大きすぎてページの境界に収まりません。

```
# ilbadm show-stats -v 1
```

次のコマンドは、規則 r1 の統計情報を 1 秒間隔で合計 3 回表示します。

```
# ilbadm show-stats -A -r r1 1 3
PKT_P   BYTES_P   PKT_U   BYTES_U   PKT_D   BYTES_D
0       0         0       0         4       196
0       0         0       0         4       196
0       0         0       0         4       196
```

次のコマンドは、規則 r1 の統計情報をそのサーバーごとに項目化し、1 秒間隔で合計 3 回表示します。

```
# ilbadm show-stats -A -r r1 -i 1 3
SERVERNAME          PKT_P   BYTES_P
_sg1.0              0       0
_sg1.1              0       0
_sg1.2              0       0
_sg1.0              0       0
```

例 10 統計情報の表示 (続き)

```
_sg1.1              0       0
_sg1.2              0       0
_sg1.0              0       0
_sg1.1              0       0
_sg1.2              0       0
```

次のコマンドは、サーバー _sg1.0 の統計情報を項目化し、タイムスタンプを付加して、1 秒間隔で合計 3 回表示します。

```
# ilbadm show-stats -A -s _sg1.0 -it 1 3
RULENAME            PKT_P   BYTES_P   TIME
r1                  0       0         2009-07-20:16.10.20
r1                  0       0         2009-07-20:16.10.21
r1                  0       0         2009-07-20:16.10.22
```

次のコマンドは、特定のオプションフィールドを指定して、統計情報を 1 秒間隔で合計 3 回表示します。

```
# ilbadm show-stats -o BYTES_D,TIME 1 3
BYTES_D   TIME
196       2009-07-20:16.14.25
0         2009-07-20:16.14.26
0         2009-07-20:16.14.27
```

例 11 健全性検査結果の表示

次のコマンドは、健全性検査の結果を表示します。

```
# ilbadm show-hc-result rule1
RULENAME   HCNAME     SERVERID   STATUS   FAIL LAST     NEXT     RTT
rule1      hc1        _sg1.0     dead     6    04:45:17 04:45:30 698
rule1      hc1        _sg1.1     alive    0    04:45:11 04:45:25 260
rule1      hc1        _sg1.2     unreach  6    04:45:17 04:45:30 0
```

例 12 NAT テーブルの表示

次のコマンドは、NAT テーブルを表示します。

```
# ilbadm show-nat 5
UDP: 124.106.235.150.53688>85.0.0.1.1024>>>82.0.0.39.4127>82.0.0.56.1024
UDP: 71.159.95.31.61528> 85.0.0.1.1024>>> 82.0.0.39.4146> 82.0.0.55.1024
UDP: 9.213.106.54.19787> 85.0.0.1.1024>>> 82.0.0.40.4114> 82.0.0.55.1024
UDP: 118.148.25.17.26676> 85.0.0.1.1024>>>82.0.0.40.4112> 82.0.0.56.1024
UDP: 69.219.132.153.56132>85.0.0.1.1024>>>82.0.0.39.4134> 82.0.0.55.1024
```

ilbadm の実際の出力では、読みやすくするためにスペースが挿入されます。

例 13 持続性テーブルの表示

次のコマンドは、持続性テーブルを表示します。

例 13　持続性テーブルの表示　　*(続き)*

```
# ilbadm show-persist 5
rule2: 124.106.235.150 --> 82.0.0.56
rule3: 71.159.95.31 --> 82.0.0.55
rule3: 9.213.106.54 --> 82.0.0.55
rule1: 118.148.25.17 --> 82.0.0.56
rule2: 69.219.132.153 --> 82.0.0.55
```

例 14　サーバーグループの表示

次のコマンドは、サーバーグループに関する基本情報を表示します。

```
# ilbadm show-servergroup
sg1: id:sg1.2 35.0.0.4:80
sg1: id:sg1.1 35.0.0.3:80
sg1: id:sg1.0 35.0.0.2:80
sg2: id:sg2.3 35.0.0.5:81
sg2: id:sg2.2 35.0.0.4:81
sg2: id:sg2.1 35.0.0.3:81
sg2: id:sg2.0 35.0.0.2:81
```

次のコマンドは、サーバーグループに関する利用可能なすべての情報を表示します。

```
# ilbadm show-servergroup -o all
sgname         serverID            minport maxport IP_address
sg1            _sg1.0              --      --      1.1.1.1
sg2            _sg2.1              --      --      1.1.1.6
sg3            _sg3.0              9001  9001      1.1.1.1
sg3            _sg3.1              9001  9001      1.1.1.2
sg3            _sg3.2              9001  9001      1.1.1.3
sg3            _sg3.3              9001  9001      1.1.1.4
sg3            _sg3.4              9001  9001      1.1.1.5
sg3            _sg3.5              9001  9001      1.1.1.6
sg3            _sg3.6              9001  9001      1.1.1.11
sg3            _sg3.7              9001  9001      1.1.1.12
sg3            _sg3.8              9001  9001      1.1.1.13
sg3            _sg3.9              9001  9001      1.1.1.14
sg3            _sg3.10             9001  9001      1.1.1.15
sg3            _sg3.11             9001  9001      1.1.1.16
sg4            _sg4.0              9001  9006      1.1.1.1
sg4            _sg4.1              9001  9006      1.1.1.6
```

例 15　規則に関連付けられたサーバーの一覧表示

次のコマンドは、規則に関連付けられているサーバーを一覧表示します。

```
# ilbadm show-server r1
SERVERID            ADDRESS         PORT RULENAME     STATUS SERVERGROUP
_sg1.0              35.0.0.10       80   rule1        E      sg1
```

例 15 規則に関連付けられたサーバーの一覧表示 (続き)

```
_sg1.1              35.0.0.11       80   rule1        E      sg1
_sg1.2              35.0.0.12       80   rule1        D      sg1
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/load-balancer/ilb
インタフェースの安定性	確実

関連項目

ilbd(1M), vmstat(1M), attributes(5)

名前

ilbd – ILB デーモン

形式

/usr/lib/inet/ilbd

機能説明

統合ロードバランサ (ILB) デーモン ilbd は、ILB のユーザーレベルのサービスを提供します。サービス管理機能 (smf(5) を参照) によって起動されるシステムデーモンです。障害管理リソース識別子 (FMRI) は次のとおりです。

svc:/network/loadbalancer/ilb:default

ilbd はコンソリデーション専用の非公開インタフェースです。attributes(5) を参照してください。

ilbd では管理インタフェースは提供されません。ILB の管理はすべて ilbadm(1M) またはプログラミングライブラリ libilb(3LIB) を介して行うようにしてください。

オプション

ilbd デーモンにオプションはありません。

使用例

例 1　ILB サービスの有効化

次のコマンドは、ILB サービスを有効にします。

```
# svcadm enable svc:/network/loadbalancer/ilb:default
```

例 2　ILB サービスの無効化

次のコマンドは、ILB サービスを無効にします。

```
# svcadm disable svc:/network/loadbalancer/ilb:default
```

エラー

ilbd デーモンは syslog(3C) を使用してステータスおよびエラーメッセージを報告します。メッセージはすべて LOG_DAEMON 機能によってログに記録されます。エラーメッセージは LOG_ERR および LOG_NOTICE 優先順位で記録され、情報メッセージは LOG_INFO 優先順位で記録されます。/etc/syslog.conf ファイルのデフォルトのエントリでは、ilbd デーモンのすべてのエラーメッセージが /var/adm/messages ログに記録されます。

ファイル

/usr/lib/inet/ilbd
　ILB デーモンのバイナリ

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/load-balancer/ilb
インタフェースの安定性	プロジェクト非公開

関連項目

svcs(1), ilbadm(1M), svcadm(1M), syslog(3C), libilb(3LIB), syslog.conf(4), attributes(5), smf(5)

注意事項

ilbd サービスは、サービス管理機能 smf(5) によって、次の障害管理リソース識別子 (FMRI) の下で管理されます。

```
svc:/network/loadbalancer/ilb:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。サービスを管理するには、承認 solaris.smf.manage.ilb が付与されている必要があります。または、権利プロファイル「ILB Management」を付与する方法もあります。

ILB サービスが有効になっていないと、ilbadm(1M) コマンドおよびほかの libilb(3LIB) クライアントアプリケーションが正しく機能しません。

名前

ilomconfig – ILOM LAN configuration utility

形式

`ilomconfig list` *subcommand* [*options*]

`ilomconfig modify` *subcommand* [*options*]

`ilomconfig enable` *subcommand* [*options*]

`ilomconfig disable` *subcommand* [*options*]

機能説明

`ilomconfig` is an Oracle Integrated Lights Out Management (ILOM) configuration utility that allows an administrator to configure the LAN interface between the Service Processor (SP) and host. The administrator can view the LAN configuration settings by means of the `list` command and configure the interface using the `enable`, `disable`, and `modify` commands.

サブコマンド

The following subcommands are available under the `list` command:

`system-summary`
List product summary information and ILOM version.

`interconnect`
List Host-to-ILOM interconnect settings.

The following subcommands are available under the `modify` command:

`interconnect`
Modifies Host-to-ILOM interconnect settings.

The following subcommand is available under the `enable` command:

`interconnect`
Enables the Host-to-ILOM interconnect.

The following subcommand is available under the `disable` command:

`interconnect`
Disables the Host-to-ILOM interconnect.

オプション

The following options are available for the `enable` and `modify` commands:

`-ipaddress=`*ipaddress*
Set the IP address for the SP side of the LAN interconnect.

`-hostipaddress=`*ipaddress*
Set the IP address for the host side of the LAN interconnect.

`-netmask=`*netmask*
Set the network mask for the IP addresses used for the LAN.

The following are general options:

`-?`, `-h`, `--help`
Display a brief usage message and then exit.

`-q`, `--quiet`
Suppress informational message output and return only error codes.

`-V`, `--version`
Display version information for `ilomconfig` and then exit.

ファイル

`/var/log/ilomconfig/ilomconfig.log`
`ilomconfig`-specific log file.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/ilomconfig
Interface Stability	Committed

関連項目

`attributes(5)`

名前

imqadmin – launch the Message Queue administration console

形式

```
/usr/bin/imqadmin [-javahome path]
/usr/bin/imqadmin -h
/usr/bin/imqadmin -v
```

機能説明

`imqadmin` launches the graphical user interface application that performs most Message Queue administration tasks. These tasks include managing broker instances (including physical destinations) and administered objects.

オプション

The following options are supported:

`-h`	Display usage help. The application is not launched.
`-javahome` *path*	Specify a path to an alternate Java 2 compatible runtime.
`-v`	Display version information.

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME`	Specify the Java 2 compatible runtime. When this environment variable is not set it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0`	Successful completion.
`>0`	An error occurred.

ファイル

`$HOME/.imq/admin/brokerlist.properties`
Contains user settings, a list of broker instances being managed.

`$HOME/.imq/admin/objectstorelist.properties`
Contains user settings, a list of object stores being managed.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

`imqbrokerd(1M)`, `imqcmd(1M)`, `imqdbmgr(1M)`, `imqkeytool(1M)`, `imqobjmgr(1M)`, `imqusermgr(1M)`, `attributes(5)`

Sun Java System Message Queue Administrator's Guide

名前

imqbrokerd – start a Message Queue broker instance

形式

```
/usr/bin/imqbrokerd [option...]
```

```
/usr/bin/imqbrokerd -h
```

機能説明

`imqbrokerd` starts an instance of the Message Queue broker. The Message Queue broker is the main component of a Message Queue message server. The broker performs reliable delivery of messages to and from Java Message Service (JMS) clients.

`imqbrokerd` uses command line options to specify broker configuration properties.

オプション

The following options are supported:

`-backup` *fileName*	Back up a Master Broker's configuration change record to *fileName*. This option only applies to broker clusters.
`-cluster` *brokerList*	Specify the list of broker instances which are connected in a cluster. This list is merged with the list in the imq.cluster.brokerlist property. This option only applies to broker clusters. *brokerList* is a comma-separated list of broker instances, each specified by *hostName*:*port* (the host on which the broker instance is running and the port number it is using) If you don't specify a value for *hostName*, `localhost` is used. If you don't specify a value for *port*, the value of `7676` is used. For example: `host1:8899,host2,:7878`.
`-dbpassword` *password*	Specify the password for a plugged-in JDBC-compliant database used as a Message Queue data store.
`-dbuser` *userName*	Specify the user name for a plugged-in JDBC-compliant data store.
`-D`*property*`-=`*value*	Set the specified broker configuration property to the *value*. The system does not validate either the configuration property or *value*. Therefore, spelling, formatting, and case is important. Message Queue can not set incorrect values passed using the `-D` option.
`-force`	Perform action without user confirmation. This option only applies when you use the `-remove` *instance* option, which normally requires confirmation.

`-h`	Display usage help. Execute nothing else on the command line.
`-javahome` *path*	Specify the path to an alternate Java 2-compatible Java Development Kit (JDK) or Java Runtime Environment (JRE) The default is to use the runtime bundled with the operating system.
`-ldappassword` *password*	Specify the password for accessing an LDAP user repository when using an LDAP server (as opposed to a built-in flat-file repository) to authenticate users of a Message Queue message server.
`-license` [*name*]	Specify the license to load, if different from the default for your Message Queue product edition. If you don't specify a license name, this lists all licenses installed on the system. Depending on the installed Message Queue edition, the values for *name* are `pe` (Platform Edition-basic features), `try` (Platform Edition-90-day trial enterprise features), and `unl` (Enterprise Edition).
`-loglevel` *level*	Specify the logging level. Valid values for *level* are `NONE`, `ERROR`, `WARNING`, or `INFO`. The default value is `INFO`.
`-metrics` *int*	Report metrics at a specific interval. Specify *int* as the number of seconds.
`-name` *brokerName*	Specify the instance name of this broker and use the corresponding instance configuration file. If you do not specify a broker name, the name of the file is set to `imqbroker`. If you run more than one instance of a broker on the same host, each must have a unique name.
`-passfile` *filename*	Specify the name of the file from which to read the passwords for the SSL keystore, LDAP user repository, or JDBC-compliant database.
`-password` *keypassword*	Specify the password for the SSL certificate keystore.
`-port` *number*	Specify the broker's Port Mapper port number. By default, this is set to `7676`. To run two instances of a broker on the same server, each broker's Port

Mapper must have a different port number. JMS clients connect to the broker instance using this port number.

`-remove` *instance*

Remove the broker instance. Delete the instance configuration file, log files, data store, and other files and directories associated with the broker instance. This option requires user confirmation unless you also specify the `-force` option.

`-reset` store|messages|durables|props

Reset the data store (or a subset of the store) or resets the configuration properties of the broker instance when the broker instance is started. The action depends on the argument provided.

store	Clear all persistent data in the data store, including messages, durable subscriptions, and transaction information store.
messages	Clear all persistent messages durable.
durables	Clear all durable subscriptions.
props	Clear all configuration information in the `config.props` instance configuration file. All properties assume default values.

`-restore` *filename*

Replace the Master Broker's configuration change record with the specified backup file. This file must have been previously created using the `-backup` option. This option only applies to broker clusters.

`-shared`

Specify that the jms connection service be implemented using the shared threadpool model, in which threads are shared among connections to increase the number of connections supported by a broker instance.

`-silent`

Turn off logging to the console.

`-tty`

Display all messages be to the console. `WARNING` and `ERROR` level messages are displayed on the console by default.

`-upgrade-store-nobackup`
: Specify that an earlier, incompatible version Message Queue data store is automatically removed when migrating to Message Queue 3.5 format.

  If you do not use this option, you must manually delete the earlier data store. This applies to both built-in (flat-file) persistence and plugged-in (JDBC-compliant) persistence. Migration of the earlier data store to a Message Queue 3.5 data store takes place the first time you start a Message Queue 3.5 broker instance on an earlier version data store.

`-version`
: Display the version number of the installed product.

`-vmargs` *are* [[*arg*]...]
: Specify arguments to pass to the Java VM. Separate arguments with spaces. If you want to pass more than one argument or if an argument contains a space, use enclosing quotation marks. For example:

```
imqbrokerd -tty -vmargs " -Xmx128m -Xincgc"
```

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME`
: Specify the Java 2 compatible runtime. When this environment variable is not set it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

ファイル

`/etc/init.d/imq`
: Shell script for starting `imqbrokerd`. This file looks at the `/etc/imq/imqbrokerd.conf` file.

`/etc/imq/imqbrokerd.conf`
: Configuration file which controls the behavior of the broker startup script.

`/etc/imq/passwd`
: Flat file user repository for authenticating users.

`/etc/imq/accesscontrol.properties`
: Controls client access to broker functionality.

`/etc/imq/passfile.sample`
: Sample passfile used by the `-passfile` option.

`/var/imq/instances/brokerName/props/config.properties`
Broker instance configuration file.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

`imqadmin(1M)`, `imqcmd(1M)`, `imqdbmgr(1M)`, `imqkeytool(1M)`, `imqobjmgr(1M)`, `imqusermgr(1M)`, `attributes(5)`

Sun Java System Message Queue Administrator's Guide

名前

imqcmd – manage Message Queue brokers

形式

```
/usr/bin/imqcmd subcommand argument [option...]
/usr/bin/imqcmd [-h | -H]
/usr/bin/imqcmd -v
```

機能説明

imqcmd manages the Message Queue broker, including resources such as connection services, physical destinations, durable subscriptions, and transactions. The utility provides a number of subcommands for managing these resources.

imqcmd supports many subcommands. Basic connection and authentication is required for the execution of every imqcmd subcommand. Use the -secure option to specify secure connections. Subcommands and their corresponding arguments and options follow the imqcmd command on the command line. See USAGE and OPTIONS.

オプション

The following options are supported:

-b *hostName*:*port*	Specify the name of the host on which the broker instance is running and the port number it is using. The default value is localhost:7676. If you do not specify the -b option, imqcmd uses the default. To specify port only, use: -b :7878. This is equivalent to -b localhost:7878 To specify name only, use: -b somehost. This is equivalent to -b somehost:7676.
-c *clientID*	Specify the ID of the durable subscriber to a topic.
-d *topicName*	Specify the name of the topic. Use this option with the list dur and destroy dur subcommands.
-f	Perform action without user confirmation. Use this option with any subcommand.
-h	Display usage help. Execute nothing else on the command line.
-H	Display usage help, attribute list, and examples. Execute nothing else on the command line.
-int *interval*	Specify the interval, in seconds, at which the metrics bkr, metrics dst, and metrics svc subcommands display metrics output. Use this option with the metrics subcommand.
-javahome	Specify an alternate Java 2 compatible runtime to use.

`-m` *metricType*

Specify the type of metric information to diplay.

Use this option with the `metrics bkr`, `metrics dst`, and `metrics svc` subcommands. The value of *metricType* depends on whether the metrics are generated for a destination, a service, or a broker.

Use one of the following values to specify *metricType*:

`ttl` Total of messages in and out of the broker (default)

`rts` Provides the same information as `ttl`, but specifies the number of messages per second

`cxn` Connections, virtual memory heap, threads

The following command displays connection, VM heap, and threads metric information for the default broker instance (`localhost:7676`) every five seconds:

```
imqcmd metrics bkr -m cxn -int 5
```

`-msp` *numSamples*

Specify the number of samples the `metrics bkr`, `metrics dst`, and `metrics svc` subcommands display in the metrics output.

`-n` *argumentName*

Specify the name of the subcommand argument. Depending on the subcommand, this might be the name of a service, a physical destination, a durable subscription, or a transaction ID.

`-o` *attribute=value*

Specify the value of an attribute. Depending on the subcommand argument, this might be the attribute of a broker, service, or destination.

`-p` *password*

Specify the administrator password.

This option is deprecated. Use the `-passfile` option instead.

`-passfile`

Specify the administrator password.

`-pst` *pauseType*

Specify whether producers, consumers, or both are paused when pausing a destination.

Use this option with the `pause dst` subcommand. Use one of the following values:

CONSUMERS Pause delivery of messages to consumers.

PRODUCERS Pause delivery of messages from producers.

ALL Pause delivery of messages to consumers and from producers.

If the `-pst` option is not specified, pauses both consumers and producers (the equivalent of `-pst ALL`).

`-rtm` *timeout*
: Specify the timeout period in seconds of an `imqcmd` subcommand. The default value is `10`.

`-rtr` *numRetries*
: Specify the number of retries attempted after an `imqcmd` subcommand times out.

  The default value is `5`.

`-s`
: Silent mode. No output is displayed.

  Use this option with any subcommand.

`-secure`
: Specify a secure administration connection to the broker instance. You must first configure the broker to enable a secure connection service.

  Use this option whenever you want a secure communication with the broker.

`-svn` *serviceName*
: Specify the service for which the connections are listed.

  Use this option with the `list cxn` subcommand.

`-t` *destinationType*
: Specify the type of a destination: `t` (topic) or `q` (queue).

`-tmp`
: Include temporary destinations when listing destinations using the list `dst` subcommand.

`-u` *name*
: Specify the administrator user name.

  If you omit this value, you are prompted for it.

`-v`
: Display version information. Execute nothing else on the command line.

使用法

Subcommands and Options

The following subcommands and associated arguments and options are supported:

compact dst [`-t` *type* `-n` *destName*]
: Compact the flat-file data store for the destination of the specified type and name. If no type and name are specified, all destinations are compacted. Destinations must be paused before they can be compacted.

commit txn `-n` *transaction_id*
: Commit the specified transaction

create dst `-t` *destinationType* `-n` *destName* [`-o` *attribute=value*] [`-o` *attribute=value1*]...
Create a destination of the specified type, with the specified name, and the specified attributes. Destination names must contain only alphanumeric characters (no spaces) and can begin with an alphabetic character or the underscore character (_).

destroy dst `-t` *destinationType* `-n` *destName*
Destroy the destination of the specified type and name.

destroy dur `-n` *subscrName* `-c` *client_id*
Destroy the specified durable subscription for the specified Client Identifier.

list cxn [`-svn` *serviceName*] [`-b` *hostName:port*]
List all connections of the specified service name on the default broker or on a broker at the specified host and port. If the service name is not specified, all connections are listed.

list dst [`-tmp`]
List all destinations, with option of listing temporary destinations as well .

list dur `-d` *destination*
List all durable subscriptions for the specified destination.

list svc
List all connection services on the broker instance.

list txn
List all transactions, being tracked by the broker.

metrics bkr [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]
Display broker metrics for the broker instance.

Use the `-m` option to specify the type of metric to display. Use one of the following values to specify *metricType*:

`ttl` Specifies the total of messages in and out of the broker (default) .

`rts` Provides the same information as `ttl`, but specifies the number of messages per second.

`cxn` Connections, virtual memory heap, threads.

Use the `-int` option to specify the interval (in seconds) at which to display the metrics. The default is 5 seconds.

Use the `-msp` option to specify the number of samples displayed in the output. A value of -1 means an unlimited number. The default value is -1.

metrics dst `-t` *type* `-n` *destName* [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]
Displays metrics information for the destination of the specified type and name.

Use the `-m` option to specify the type of metrics to display. Use one of the following values to specify *metricType*:

`ttl` Specifies the number of messages flowing in and out of the broker and residing in memory.

`rts` Provides the same information as `ttl`, but specifies the number of messages per second.

`con` Displays consumer related metrics.

`dsk` Displays disk usage metrics.

Use the `-int` option to specify the interval (in seconds) at which to display the metrics. The default is 5 seconds.

Use the `-msp` option to specify the number of samples displayed in the output. A value of -1 means an unlimited number. The default value is 5.

metrics svc `-n` *serviceName* [`-m` *metricType*] [`-int` *interval*] [`-msp` *numSamples*]
List metrics for the specified service on the broker instance. Use the `-m` option to specify the type of metric to display. Use one of the following values to specify *metricType*:

`ttl` Total of messages in and out of the broker (default)

`rts` Provides the same information as `ttl`, but specifies the number of messages per second

`cxn` Connections, virtual memory heap, threads

Use the `-int` option to specify the interval (in seconds) at which to display the metrics. The default is 5 seconds.

Use the `-msp` option to specify the number of samples displayed in the output. A value of -1 means an unlimited number. The default value is -1.

pause bkr
Pause the broker instance.

pause dst [`-t` *type* `-n` *destName*] [`-pst` *pauseType*]
Pause the delivery of messages to consumers (`-pst CONSUMERS`), or from producers (`-pst PRODUCERS`), or both (`-pst ALL`), for the destination of the specified type and name. If no destination type or name are specified, all destinations are paused.

pause svc `-n` *serviceName*
Pause the specified service running on the broker instance. You cannot pause the administative service.

purge dst `-t` *destinationType* `-n` *destName*
Purge messages at the destination with the specified type and name.

purge dur `-n` *subscrName* `-c` *client_id*
Purge all messages for the specified client identifier.

query bkr
List the current settings of properties of the broker instance. Show the list of running brokers (in a multi-broker cluster) that are connected to the specified broker.

query dst `-t` *destinationType* `-n` *destName*
List information about the destination of the specified type and name.

query svc `-n` *serviceName*
Display information about the specified service running on the broker instance.

query txn `-n` *transaction_id*
List information about the specified transaction.

reload cls
Forces all the brokers in a cluster to reload the `imq.cluster.brokerlist` property and update cluster information. This subcommand only applies to broker clusters.

restart bkr
Shut down and restart the broker instance. This command restarts the broker using the options specified when the broker was first started. If you want different options to be in effect, you must shut down the broker and then start it again, specifying the options you want.

resume bkr
Resume the broker instance.

resume dst [`-t` *type*] [`-n -destName`]
Resumes the delivery of messages for the paused destination of the specified type and name. If no destination type and name are specified, all destinations are resumed.

resume svc `-n` *serviceName*
Resume the specified service running on the broker instance.

rollback txn `-n` *transaction_id*
Roll back the specified transaction.

shutdown bkr
Shut down the broker instance

update bkr `-o` *attribute=value* [`-o` *attribute=value*]...
Change the specified attributes for the broker instance.

update dst `-t` *destinationType* `-n` *destName* `-o` *attribute=value* [`-o` *attribute=value1*]...
Update the value of the specified attributes at the specified destination..

update svc `-n` *serviceName* `-o` *attribute=value* [`-o` *attribute=value1*]...
Update the specified attribute of the specified service running on the broker instance.

Attribute Value Pairs

You can specify attributes with the create and update subcommands. Applicable attributes depend on the subcommand arguments.

The following attributes are supported:

Queue (dst):

maxTotalMsgBytes
: Value: Integer (maximum total size of messages, in bytes)

  Default: `0` (unlimited)

maxBytesPerMsg
: Value: Integer (maximum size of a single message, in bytes)

  Default: `0` (unlimited)

maxNumMsgs
: Value: Integer (maximum total number of messages)

  Default: `0` (unlimited)

consumerFlowLimit
: Value: Integer Initial number of queued messages sent to active consumers before load-balancing starts A value of `-1` means an unlimited number.

  Default: `1000`

isLocalOnly
: Value: Boolean (destination limited to delivering messages to local consumers only) Default: `false`

limitBehavior
: Value: Specify how broker responds when memory-limit is reached. Use one of the following values:

FLOW_CONTROL	Slows down producers
REMOVE_OLDEST	Purges oldest messages
REJECT_NEWEST	Rejects the newest messages

  Default: REJECT_NEWEST

localDeliveryPreferred
: Value: Boolean Specify messages be delivered to remote consumers only if there are no consumers on the local broker. Requires that the destination not be restricted to local-only delivery (`isLocalOnly = false`)

  Default: `false`

maxNumActiveConsumers
: Value: Integer (maximum number of active consumers in load-balanced delivery) A value of `-1` means an unlimited number.

  Default: `1`

maxNumBackupConsumers
: Value: Integer (maximum number of backup consumers in load-balanced delivery) A value of `-1` means an unlimited number.

  Default: `0`

maxNumProducers
: Value: (maximum total number of producers) A value of `-1` means an unlimited number.

  Default: `-1`

useDMQ
: Specify whether a destination's dead messages are discarded or put on the dead message queue.

  Default: `true`

Topic (dst):

consumerFlowLimit
: Value: Integer Maximum number of messages delivered to a consumer in a single batch. A value of `-1` means an unlimited number.

  Default: `1000`

isLocalOnly
: Value: Boolean (destination limited to delivering messages to local consumers only)

  Default: `false`

limitBehavior
: Value: Specify how broker responds when memory-limit is reached. Use one of the following values:

FLOW_CONTROL	Slows down producers
REMOVE_OLDEST	Purges the oldest messages
REJECT_NEWEST	Rejects the newest messages

  Default: REJECT_NEWEST

maxBytesPerMsg
: Value: Integer (maximum size of a single message, in bytes)

  Default: `0` (unlimited)

maxNumMsgs
: Value: Integer (maximum total number of messages) A value of `-1` means an unlimited number.

  Default: `-1`

maxNumProducers
: Value: (maximum total number of producers)

  Default: `0` (unlimited)

maxTotalMsgBytes
: Value: Integer (maximum total size of messages, in bytes) A value of `-1` means an unlimited number.

  Default: `-1`

useDMQ

Specify whether a destination's dead messages are discarded or put on the dead message queue.

Default: `true`

Broker (bkr):

imq.autocreate.destination.useDMQ

Value: Boolean. Set the `useDMQ` attribute to `true` to enable all autocreated physical destinations on a broker to use the dead message queue. Set the `useDMQ` attribute to `false` to disable all autocreated physical destinations on a broker from using the dead message queue.

Default: `true`

imq.autocreate.queue

Value: Boolean

Default: `true`

imq.autocreate.queue.maxNumActiveConsumers

Value: Integer (maximum number of consumers that can be active in load-balanced delivery from an autocreated queue destination) A value of `-1` means an unlimited number.

Default: `1`

imq.autocreate.queue.maxNumBackupConsumers

Value: Integer (maximum number of backup consumers that can take the place of active consumers) A value of `-1` means an unlimited number.

Default: `0`

imq.autocreate.topic

Value: Boolean

Default: `true`

imq.cluster.url

Value: String (location of cluster configuration file)

Default: none

imq.log.file.rolloverbytes

Value: Integer (maximum size of a log file, in bytes)

	Default: `0` (no rollover based on size)
imq.log.file.rolloversecs	Value: Integer (maximum age of a log file, in seconds)
	Default: `0` (no rollover based on age)
imq.log.level	Value: String (`NONE`, `ERROR`, `WARNING`, `INFO`)
	Default: `INFO`
imq.message.max_size	Value: Integer (maximum size of a single message, in bytes)
	Default: `70m`
imq.portmapper.port	Value: Integer
	Default: `7676`
imq.system.max_count	Value: Integer (maximum total number of messages)
	Default: `0` (no limit)
imq.system.max_size	Value: Integer (maximum total size of messages, in bytes)
	Default: `0` (no limit)

Service (svc):

maxThreads	Value: Integer (maximum threads assigned)
	Default: Depends on service
minThreads	Value: Integer (minimum threads assigned)
	Default: Depends on service
port	Value: Integer
	Default: `0` (dynamically allocated)

使用例

例 1 Shutting Down a Broker

The following command shuts down a broker for hostname `myserver` on port `7676`:

```
mqcmd shutdown bkr -b myserver:7676
```

例 2 Restarting a Broker

The following command restarts a broker for hostname `myserver`:

```
imqcmd restart bkr -b myserver
```

例 3 Pausing a Service

The following command pauses a broker for hostname `localhost` on port `7676`, with a *serviceName* of `jms`:

```
imqcmd pause svc -n jms -b :7676
```

例 4 Resuming a Service

The following command resumes a service for hostname `localhost` on port `7676`, with a *serviceName* of `jms`:

```
imqcmd resume svc -n jms -b myserver:7676
```

例 5 Creating a Queue Destination

The following command creates a queue destination for hostname `myserver` on port `7676`, with a *destName* of `myFQ`, a *queueDeliveryPolicy* of `Failover`, and a *maxBytesPerMsg* of 10000:

```
imqcmd create dst -n myFQ -t q -o "queueDeliveryPolicy=f" \
     -o "maxBytesPerMsg=10000" -b myserver:7676
```

例 6 Purging a Queue Destination

The following command purges a queue destination for hostname `myserver` on port `7676`, with a *destName* of `myFQ`:

```
imqcmd purge dst -n myFQ -t q -b myserver:7676
```

例 7 Listing Destinations on a Broker

The following command lists destinations for hostname `myserver` on port `7676`:

```
imqcmd list dst -b myserver:7676
```

例 8 Updating a Portmapper Port

The following command updates a portmapper port on hostname `myserver` from port `7676` to `7878`:

```
imqcmd update bkr -o "imq.portmapper.port=7878"
```

例 9 Updating the Maximum Number of Messages in the Queue

The following command updates the maximum number of messages in the queue to `2000` for `myserver` on port `8080` with a *destName* of `TestQueue`:

```
imqcmd update dst -b myserver:8080 -n TestQueue -t q -o "maxNumMsgs=2000"
```

例 10 Updating the Maximum Threads

The following command updates the maximum threads `jms` connection service to `200` for hostname `localhost` on port `7676`:

```
imqcmd update svc -n jms -o "minThreads=200"
```

例 11 Listing Durable Subscriptions

The following command lists durable subscriptions for a topic with hostname `localhost` on port `7676` with a *destName* of `myTopic`:

```
imqcmd list dur -d myTopic
```

例 12 Destroying Durable Subscriptions

The following command destroys subscriptions for hostname `localhost` on port `7676` with a *dursubName* of `myDurSub` and a *client_ID* of `111.222.333.444`:

```
imqcmd destroy dur -n myDurSub -c "111.222.333.444"
```

例 13 Listing All Transactions

The following command lists all transactions on a broker with hostname `localhost` on port `7676`:

```
imqcmd list txn
```

例 14 Displaying Information About a Transaction

The following command displays information about a transaction with hostname `localhost` on port `7676`, and a *transactionID* of `1234567890`

```
imqcmd query txn -n 1234567890
```

例 15 Committing a Transaction

The following command commits a transaction with hostname `localhost` on port `7676`, and a *transactionID* of `1234567890`:

```
imqcmd commit txn -n 1234567890
```

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME` Specify the Java 2 compatible runtime. When this environment variable is not set it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

imqadmin(1M), imqbrokerd(1M), imqdbmgr(1M), imqkeytool(1M), imqobjmgr(1M), imqusermgr(1M), attributes(5)

Sun Java System Message Queue Administrator's Guide

名前

imqdbmgr – manage a plugged-in JDBC-compliant Message Queue data store

形式

`/usr/bin/imqdbmgr` *subcommand argument* `[ [`*option*`...]]`

`/usr/bin/imqdbmgr -h | -help`

`/usr/bin/imqdbmgr -v | -version`

機能説明

The `imqdbmgr` utility creates and manages a Java DataBase Connectivity (JDBC) compliant database used for Message Queue persistent storage.

The database can be either embedded or external. To use a JDBC-compliant database (and the `imdbmgr` utility), you need to first set a number of JDBC-related properties in the broker instance configuration file. See the Sun Java System Message Queue Administrator's Guide for additional information.

`imqdbmgr` supports four management subcommands. These *subcommand*s, and their corresponding *argument*s and *option*s follow the `imqdbmgr` command on the command line. See `USAGE` and `OPTIONS`.

The following subcommands are supported:

create	Create a Message Queue database schema.
delete	Delete Message Queue database tables in the current data store.
recreate	Delete Message Queue database tables and recreate Message Queue database schema in the curent data store.
reset	Reset the database table lock to allow other processes to access database tables.

The `imqdbmgr` subcommands support the following arguments:

all	Indicates the subcommand applies to the data store, as well as the database tables.
lck	Indicates the subcommand applies to the database table lock.
oldtbl	Indicates the subcommand applies to an older version of the database tables.
tbl	Indicates the subcommand applies to the database tables only.

オプション

The following options are supported:

`-b` *brokerName*	Specify the broker instance name and corresponding instance configuration properties. If this option is not specified, the default broker instance is assumed. Use this option with the `create`, `delete`, `recreate`, or `reset` subcommands.
`-D` *property=value*	Set system property *property* to *value*.

	Use this option with the `create`, `delete`, `recreate`, or `reset` subcommands.
`-h` | `-help`	Display usage help. Execute nothing else on the command line.
`-p` *password*	Specify the database password. Use this option with the `create`, `delete`, `recreate`, or `reset` subcommands.
`-u` *userName*	Specify the database user name. Use this option with the `create`, `delete`, `recreate`, or `reset` subcommands.
`-v` | `-version`	Display version information. Execute nothing else on the command line.

使用法

The following subcommands and associated arguments are supported:

create all	Create a new embedded data store and Message Queue database schema for a specified or default broker instance.
create tbl [`-u` *userName*] [`-p` *password*]	Create Message Queue database schema in an existing data store for a specified or default broker instance.
delete tbl [`-u` *userName*] [`-p` *password*]	Delete Message Queue database tables in the current data store for a specified or default broker instance.
delete oldtbl [`-u` *userName*] [`-p` *password*]	Delete the earlier version of Message Queue database tables. Used after the data store has been automatically migrated to the current version of Message Queue.
recreate tbl [`-u` *userName*] [`-p` *password*]	Delete Message Queue database tables and recreate Message Queue database schema in the current data store for a specified or default broker instance.
reset lck	Reset the database table lock to allow other processes to access database tables.

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME`	Specify the Java 2 compatible runtime. When this environment variable is not set it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/var/imq/instances/`*brokerName*`/dbstore` Recommended directory in which to create an embedded database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

`imqadmin(1M)`, `imqbrokerd(1M)`, `imqcmd(1M)`, `imqusermgr(1M)`, `imqkeytool(1M)`, `imqobjmgr(1M)`, `attributes(5)`

Sun Java System Message Queue Administrator's Guide

名前

imqkeytool – generate a self-signed certificate for secure communication

形式

`/usr/bin/imqkeytool [-broker] [-servlet` *keystore_location*`]`

`/usr/bin/imqkeytool -h`

機能説明

The `imqkeytool` utility generates a self-signed certificate for secure communication. The certificate can be used by a broker instance to establish a secure connection with a client, or by a Message Queue-supplied HTTPS servlet to establish a secure connection with a broker instance. An HTTPS servlet is an SSL-enabled varient of the HyperText Transfer Protocol that establishes a secure connection with a broker instance.

Without an option, `imqkeytool` generates a self-signed certificate for a broker instance.

`imqkeytool` uses command line options to specify whether the certificate is used by a broker instance or by a servlet.

オプション

The following options are supported:

`-broker`	Generate a self-signed certificate for the broker and places it in the Message Queue keystore. All broker instances running on a system must use the same certificate.
`-h`	Display usage help. Do not execute anything else on the command line.
`-servlet` *keystore_location*	Generate a self-signed certificate for an HTTPS servlet and places it in *keystore_location*. *keystore_location* refers to the location of the keystore. You should move this keystore to a location where it is accessible and readable by the Message Queue HTTPS servlet to establish a secure connection with a broker.

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME`	Specify the Java 2 compatible runtime. When this environment variable is not set it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0`	Successful completion.
`>0`	An error occurred.

ファイル

`/etc/imq/keystore`	Contains Message Queue keystore in which `imqkeytool` stores a self-signed certificate for brokers.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

imqadmin(1M), imqbrokerd(1M), imqcmd(1M), imqdbmgr(1M), imqobjmgr(1M), imqusermgr(1M), attributes(5)

Sun Java System Message Queue Administrator's Guide

名前

imqobjmgr – manage Message Queue administered objects

形式

```
/usr/bin/imqobjmgr subcommand [[option]...]
/usr/bin/imqobjmgr -i fileName
/usr/bin/imqobjmgr -h | [-H] | -help | -Help
/usr/bin/imqobjmgr -v
```

機能説明

`imqobjmgr` manages Message Queue administered objects in an object store accessible using JNDI. Administered objects allow JMS clients to be provider-independent by insulating them from provider-specific naming and configuration formats.

`imqobjmgr` supports five management subcommands. These *subcommand*s, and their corresponding *option*s follow the `imqobjmgr` command on the command line. See `USAGE` and `OPTIONS`.

The following subcommands are supported:

add	Add a new administered object
delete	Delete an administered object
list	Display a list of administered objects
query	Display information about administered objects
update	Update administered objects

You can use the `-i` option to specify the name of an input file that uses java property file syntax to represent all or part of any `imqobjmgr` subcommand clause. The `-f`, `-s`, and `-pre` options can be used with any `imqobjmgr` subcommand.

オプション

The following options are supported:

`-f`	Perform action without user confirmation.
`-h` | `-help`	Display usage help. Execute nothing else on the command line.
`-H` | `-Help`	Display usage help, attribute list, and examples. Execute nothing else on the command line.
`-i` *fileName*	Specify the name of an input file containing all or part of the subcommand clause, specifying object type, lookup name, object attributes, object store attributes, or other options. Use this option for repetitive information, such as object store attributes.
`-j` *attribute=value*	Specify attributes necessary to identify and access a JNDI object store.
`-javahome`	Specify an alternate Java 2 compatible runtime to use. `imqobjmgr` uses the runtime bundled with the operating system by default.

`-l` *lookupName*	Specify the JNDI lookup name of an administered object. This name must be unique in the object store's context.
`-o` *attribute=value*	Specify the attributes of an administered object.
`-pre`	Run command in preview mode. Preview mode indicates what will be done without performing the command.
`-r` *read-only_state*	Specify if an administered object is a read-only object. A value of true indicates the administered object is a read-only object. JMS clients cannot modify the attributes of read-only administered objects. The read-only state is set to false by default.
`-s`	Silent mode. No output is displayed.
`-t` *type*	Specify the type of an administered object:

```
q = queue
t = topic
cf = ConnectionFactory
qf = queueConnectionFactory
tf = topicConnectionFactory
xcf = XA ConnectionFactory (distributed transactions)
xqf = XA queueConnectionFactory (distributed transactions)
xtf = XA topicConnectionFactory (distributed transactions)
e = SOAP endpoint (used to support SOAP messaging)
```

`-v`	Display version information. Execute nothing else on the command line.

使用法

This section provides information on subcommands, options, and attribute value pairs.

Subcommands and Options

The following subcommands and corresponding options are supported:

add `-t` *type* `-l` *lookupName* [`-o` *attribute=value*]... `- j` *attribute=value*...
: Add a new administered object of the specified type, lookup name , and object attributes to an object store.

delete `-t` *type* `-l` *lookupName* `-j` *attribute=value*...
: Delete an administered object, of the specified type and lookup name from an object store.

list [`-t` *type*] `-j` *attribute=value*...
: Display a list of administered objects of a specified type, or all administered objects, in an object store.

query `-l` *lookupName* `-j` *attribute=value*...
: Display information about an administered object of a specified lookup name in an object store.

update -l *lookupName* [-o *attribute=value*]... -j *attribute=value*...
Update the specified attribute values of an administered object of the specified lookup name in an object store.

Attribute Value Pairs

The following attribute value pairs are supported for the specified administered object types:

Type = ConnectionFactories: ConnectionFactory, TopicConnectionFactory, QueueConnectionFactory, XAConnectionFactory, XATopicConnectionFactory, and XAQueueConnectionFactory

imqAckOnAcknowledge
Value: String (`true`, `false`, not specified)

Default: not specified

imqAckOnProduce
Value: String (`true`, `false`, not specified)

Default: not specified

imqAckTimeout
Value: String (time in milliseconds)

Default: `0` (no timeout)

imqAddressList
Value: String

Default: not specified

imqAddressListBehavior
Value: String

Default: Priority

imqAddressListIterations
Value: Integer

Default: 1

imqBrokerHostName
Used if imqConnectionType is `TCP` or `TLS`. This attribute type is only supported in Message Queue 3.0.

Value: String

Default:`localhost`

imqBrokerHostPort
Used if imqConnectionType is `TCP` or `TLS`. This attribute type is only supported in Message Queue 3.0.

Value: Integer

Default: `7676`

imqBrokerServicePort
Used if imqConnectionType is `TCP` or `TLS`. This attribute type is only supported in Message Queue 3.0.

Value: Integer

Default: `0`

imqConfiguredClientID
Value: String (ID number)

Default: no ID specified

imqConnectionFlowCount
Value: Integer

Default: 100

imqConnectionFlowLimit
Value: Integer

Default: 1000

imqConnectionFlowLimitEnabled
Value: Boolean

Default: `false`

imqConnectionType
This attribute type is only supported in Message Queue 3.0.

Value: String (`TCP`, `TLS`, `HTTP`).

Default: `TCP`

imqConnectionURL
Used if imqConnectionType is `HTTP`. This attribute type is only supported in Message Queue 3.0.

Value: String

Default: `http://`*`localhost`*`/imq/tunnel`

imqConsumerFlowLimit
Value: Integer

Default: 1000

imqConsumerFlowThreshold
: Value: Integer

  Default: 50

imqDefaultPassword
: Value: String

  Default: `guest`

imqDefaultUsername
: Value: String

  Default: `guest`

imqDisableSetClientID
: Value: Boolean

  Default: `false`

imqJMSDeliveryMode
: Value: Integer (`1`=non-persistent, `2`=persistent)

  Default: `2`

imqJMSExpiration
: Value: Long (time in milliseconds)

  Default: `0` (does not expire)

imqJMSPriority
: Value: Integer (`0` to `9`)

  Default: `4`

imqLoadMaxToServerSession
: Value: Boolean

  Default: `true`

imqOverrideJMSDeliveryMode
: Value: Boolean

  Default: `false`

imqOverrideJMSExpiration
: Value: Boolean

  Default: `false`

imqOverrideJMSHeadersToTemporaryDestinations
: Value: Boolean

  Default: `false`

imqOverrideJMSPriority
Value: Boolean

Default: `false`

imqQueueBrowserMaxMessagesPerRetrieve
Value: Integer

Default: `1000`

imqBrowserRetrieveTimeout
Value: Long (time in milliseconds)

Default: `60,000`

imqReconnectAttempts
Value: Integer

Default: `0`

imqReconnectEnabled
Value: Boolean

Default: `false`

imqReconnectInterval
Value: Long (time in milliseconds)

Default: `3000`

imqSetJMSXAppID
Value: Boolean

Default: `false`

imqSetJMSXConsumerTXID
Value: Boolean

Default: `false`

imqSetJMSXProducerTXID
Value: Boolean

Default: `false`

imqSetJMSXRcvTimestamp
Value: Boolean

Default: `false`

imqSetJMSXUserID
Value: Boolean

Default: `false`

imqSSLIsHostTrusted
: Used if imqConnectionType is `TLS`. This attribute type is only supported in Message Queue 3.0.

  Value: Boolean

  Default: `true`

Type = Destinations: Topic and Queue

imqDestinationDescription
: Value: String

  Default: no description

imqDestinationName
: Value: String

  Default: `Untitled_Destination_Object`

Type = Endpoint (SOAP Endpoint)

imqEndpointDescription
: Value: String

  Default: `A description for the endpoint object`

imqEndpointName
: Value: String

  Default: `Untitled_Endpoint_Object`

imqSOAPEndpointList
: Value: String (one or more space-separated URLs)

  Default: no url

使用例

例 1　Adding a Topic Administered Object to an Object Store

Where JNDI lookup name=`myTopic` and `imqDestinationName=MyTestTopic`, the following command adds to an LDAP server object store:

```
imqobjmgr add -t t -l "cn=myTopic"\
-o "imqDestinationName=MyTestTopic"\
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"\
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
```

Where JNDI lookup name=`myTopic` and `imqDestinationName=MyTestTopic`, the following command adds to a file system object store:

```
imqobjmgr add -t -l "cn=myTopic"\
-o "imqDestinationName=MyTestTopic"\
-j \
  "java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory"\
-j "java.naming.provider.url=file:/home/foo/imq_admin_objects"
```

例 1 Adding a Topic Administered Object to an Object Store *(続き)*

Where JNDI lookup name=`myTopic` and `imqDestinationName=MyTestTopic`, the following command adds to a file system object store, using an input file:

```
imqobjmgr -i inputfile
```

The associated input file consists of the following:

```
cmdtype=add
obj.type=t
obj.lookupName=cn=myTopic
obj.attrs.imqDestinationName=MyTestTopic
objstore.attrs.java.naming.factory.initial=com.sun.jndi.fscontext.\
  RefFSContextFactory
objstore.attrs.java.naming.provider.url=file:/home/foo/imq_admin_objects
```

例 2 Adding a QueueConnectionFactory Administered Object to an Object Store

Where JNDI lookup name=`myQCF`, `read-only state=true`, `imqAddressList=mq://foohost:777/jms`, the following command adds to an LDAP server object store:

```
imqobjmgr add -t qf -l "cn=myQCF" -r true\

-o "imqAddressList=mq://foohost:777/jms"\
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"\
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"\
```

Where JNDI lookup name=`myQCF`, `read-only state=true`, `imqAddressList=mq://foohost:777/jms`, the following command adds to an LDAP server object store using an input file:

```
imqobjmgr -i inputfile
```

The associated input file consists of the following:

```
cmdtype=add
obj.type=qf
obj.lookupName=cn=myQCF
obj.readOnly=true
obj.attrs.imqAddressList=mq://foohost:777/jms
objstore.attrs.java.naming.factory.initial=com.sun.jndi.\
  ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=ldap://mydomain.com:389/o=imq
```

Where JNDI lookup name=`myQCF`, read-only state=`true`, `imqAddressList=mq://foohost:777/jms`, the following command adds to an LDAP server object store, using both an input file and command options:

例 2 Adding a QueueConnectionFactory Administered Object to an Object Store (続き)

```
imqobjmgr add -t qf -l "cn=myQCF"\
-o "imqAddressList=mq://foohost:777/jms"\
-i inputfile
```

The associated input file consists of the following:

```
objstore.attrs.java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=ldap://mydomain.com:389/o=imq
```

例 3 Deleting a Topic Administered Object from an Object Store

Where JNDI lookup name=`myTopic` and no confirmation is requested, the following command deletes from an LDAP server object store:

```
imqobjmgr delete -f -l "cn=myTopic"\
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"\
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
```

例 4 Querying Information About a Topic Administered Object

Where JNDI lookup name=`myTopic`, the following command queries from an LDAP server object store using simple authentication scheme:

```
imqobjmgr query -l "cn=myTopic"\
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"\
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"\
-j "java.naming.security.authentication=simple"\
-j "java.naming.security.principal=uid=foo,ou=imqobjmgr,o=imq"\
-j "java.naming.security.credentials=foo"
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

`imqadmin(1M)`, `imqcmd(1M)`, `imqbrokerd(1M)`, `imqkeytool(1M)`, `imqusermgr(1M)`, `attributes(5)`

Sun Java System Message Queue Administrator's Guide

名前

imqusermgr – command utility for managing a Message Queue user repository

形式

```
/usr/bin/imqusermgr subcommand [[option]...]
/usr/bin/imqusermgr -h
/usr/bin/imqusermgr -v
```

機能説明

The `imqusermgr` utility manages a file-based user repository to authenticate and authorize users of a Message Queue message server.

`imqusermgr` provides subcommands for adding, deleting, updating, and listing user entries in the repository.

`imqusermgr` supports four management subcommands. These *subcommand*s, and their corresponding *option*s follow the `imqusermgr` command on the command line. See `USAGE` and `OPTIONS`.

The following subcommands are supported:

add	Add a new user and associated password to the repository.
delete	Delete a user from the repository.
list	Display information users in the repository.
update	Update the password or state of a user in the repository.

オプション

The following options are supported:

`-a` *active_state*	Specify if user's state is active or inactive. An inactive user cannot create connections to the Message Queue message server. Valid values for *active_state* are `true` or `false`. Specify `true` for active or `false` for inactive. the default is `true`. Use this option with the `update` subcommmand.
`-f`	Perform action without user confirmation. Use this option with the `delete` and `update` subcommmands.
`-g` *group*	Specify the group of the user. Valid values for group are `admin`, `user`, and `anonymous`. Use this option with the `add` subcommmand.
`-h`	Display usage help. Exceute nothing else on the command line.
`-i` *brokerName*	Specify the broker instance user repository to which the command applied. If you do not specify *brokerName*, the default *brokerName* is assumed.

Use this option with the `add`, `delete`, `list`, and `update` subcommmands.

`-p` *password* Specify user password.

Use this option with the `add` and `update` subcommmands.

`-s` Silent mode. Display no output

Use this option with the `add`, `delete`, and `update` subcommands.

`-u` *userName* Specify user name.

userName cannot contain the following characters: asterisk (`*`), colon (`:`), NEWLINE, or RETURN.

Use this option with the `add`, `delete`, `update` and `list` subcommands.

`-v` Display version information. Exceute nothing else on the command line.

使用法

The following subcommands and corresponding options are supported:

add `-u` *userName* `-p` *password* [`-g` *group*] [`-s`] [`-i` *brokerName*]
Add a new user and associated password to the repository, and optionally specify the user's group.

delete `-u` *userName* [`-s`] [`-f`] [`-i` *brokerName*]
Delete a user from the repository.

list [`-u` *user_name*] [`-i` *brokerName*]
Display information about the specified user in the repository. If no user is specified, all users are displayed.

update `-u` *userName* `-p` *password* [`-a` *state*] [`-s`] [`-f`] [`-i` *brokerName*]
update `-u` *userName* `-a` *state* [`-p` *password*] [`-s`] [`-f`] [`-i` *brokerName*]
Update the password or state (or both) of a user.

環境変数

The following environment variables affect the execution of this command:

`IMQ_JAVAHOME` Specify the Java 2 compatible runtime. When this environment variable is not set, it defaults to `/usr/j2se`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/etc/imq/passwd` Flat-file user repository.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWiq

関連項目

imqadmin(1M), imqbrokerd(1M), imqcmd(1M), imqdbmgr(1M), imqkeytool(1M), imqobjmgr(1M), attributes(5)

Sun Java System Message Queue Administrator's Guide

名前

in.chargend – UDP or TCP character generator service daemon

形式

```
in.chargend
```

```
FMRI svc:/internet/chargen:default
```

機能説明

FMRI stands for Fault Management Resource Identifier. It is used to identify resources managed by the Fault Manager. See fmd(1M) and smf(5).

The in.chargend service provides the server-side of the character-generator protocol. This protocol is used for debugging and bandwidth measurement and is available on both TCP and UDP transports, through port 19.

The in.chargend service is an inetd(1M) smf(5) delegated service. The in.chargend detects which transport is requested by examining the socket it is passed by the inetd daemon.

TCP-based service: Once a connection is established, the in.chargend generates a stream of data. Any data received is discarded. The server generates data until the client program terminates the connection. Note that the data flow is limited by TCP flow control mechanisms.

UDP-based service: The in.chargend listens for UDP datagrams. When a datagram is received, the server generates a UDP datagram in response containing a random number of ASCII characters (ranging from 0 to 512 characters). Any received data is ignored.

The in.chargend data consists of a pattern of 72 character lines containing the printable, 7–bit ASCII characters. Each line is terminated with a carriage return and a line feed character.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/legacy-network-services
Interface Stability	Committed

関連項目

inetd(1M), attributes(5), smf(5)

RFC 864

名前

in.comsat, comsat – biff server

形式

/usr/sbin/in.comsat

機能説明

comsat is the server process which listens for reports of incoming mail and notifies users who have requested to be told when mail arrives. It is invoked as needed by inetd(1M), and times out if inactive for a few minutes.

comsat listens on a datagram port associated with the biff service specification (see services(4)) for one line messages of the form

user@mailbox-offset

If the *user* specified is logged in to the system and the associated terminal has the owner execute bit turned on (by a biff y), the *offset* is used as a seek offset into the appropriate mailbox file, and the first 7 lines or 560 characters of the message are printed on the user's terminal. Lines which appear to be part of the message header other than the From, To, Date, or Subject lines are not printed when displaying the message.

ファイル

/var/adm/utmpx　user access and administration information

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

svcs(1), inetadm(1M), inetd(1M), svcadm(1M),services(4), attributes(5), smf(5)

注意事項

The message header filtering is prone to error.

The in.comsat service is managed by the service management facility, smf(5), under the service identifier:

svc:/network/comsat:default

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

in.daytimed – UDP or TCP daytime protocol service daemon

形式

```
in.daytimed
```

```
FMRI svc:/internet/daytime:default
```

機能説明

FMRI stands for Fault Management Resource Identifier. It is used to identify resources managed by the Fault Manager. See fmd(1M) and smf(5).

The in.daytimed service provides the server-side of the daytime protocol. This protocol is used for debugging and bandwidth measurement and is available on both TCP and UDP transports, through port 13.

The in.daytimed service is an inetd(1M) smf(5) delegated service. The in.daytimed detects which transport is requested by examining the socket it is passed by the inetd daemon.

TCP-based service
: Once a connection is established, the in.daytimed generates the current date and time in ctime(3C) format as 7-bit ASCII and sends it through the connection. The server then closes the connection. Any data received from the client side of the connection is discarded.

UDP-based service
: The in.daytimed listens for UDP datagrams. When a datagram is received, the server generates the current date and time in ctime(3C) format as 7-bit ASCII and inserts it in a UDP datagram sent in response to the client's request. Any received data is ignored.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/legacy-network-services
Interface Stability	Committed

関連項目

inetd(1M), attributes(5), smf(5)

RFC 867

名前

in.dhcpd – Dynamic Host Configuration Protocol server

形式

```
/usr/lib/inet/in.dhcpd  [-denv] [-h relay_hops] [-i interface, ...]
 [-l syslog_local_facility] [-b automatic | manual]
 [-o DHCP_offer_time] [-t dhcptab_rescan_interval]
```

```
/usr/lib/inet/in.dhcpd  [-dv] [-h relay_hops] [-i interface,]...
 [-l syslog_local_facility] -r IP_address | hostname, ...
```

機能説明

`in.dhcpd` is a daemon that responds to Dynamic Host Configuration Protocol (DHCP) requests and optionally to BOOTP protocol requests. The daemon forks a copy of itself that runs as a background process. It must be run as root. The daemon has two run modes, DHCP server (with optional BOOTP compatibility mode) and BOOTP relay agent mode.

The first line in the `SYNOPSIS` section illustrates the options available in the DHCP/BOOTP server mode. The second line in the SYNOPSIS section illustrates the options available when the daemon is run in BOOTP relay agent mode.

The DHCP and BOOTP protocols are used to provide configuration parameters to Internet hosts. Client machines are allocated their IP addresses as well as other host configuration parameters through this mechanism.

The DHCP/BOOTP daemon manages two types of DHCP data tables: the `dhcptab` configuration table and the DHCP network tables.

See `dhcptab(4)` regarding the dhcptab configuration table and `dhcp_network(4)` regarding the DHCP network tables.

The `dhcptab` contains macro definitions defined using a `termcap`-like syntax which permits network administrators to define groups of DHCP configuration parameters to be returned to clients. However, a DHCP/BOOTP server always returns hostname, network broadcast address, network subnet mask, and IP maximum transfer unit (MTU) if requested by a client attached to the same network as the server machine. If those options have not been explicitly configured in the `dhcptab`, `in.dhcpd` returns reasonable default values.

The `dhcptab` is read at startup, upon receipt of a `SIGHUP` signal, or periodically as specified by the `-t` option. A `SIGHUP` (sent using the command `svcadm refresh network/dhcp-server`) causes the DHCP/BOOTP daemon to reread the `dhcptab` within an interval from `0-60` seconds (depending on where the DHCP daemon is in its polling cycle). For busy servers, users should run `svcadm restart network/dhcp-server` to force the `dhcptab` to be reread.

The DHCP network tables contain mappings of client identifiers to IP addresses. These tables are named after the network they support and the datastore used to maintain them.

The DHCP network tables are consulted during runtime. A client request received from a network for which no DHCP network table exists is ignored.

This daemon is obsolete and is subject to removal in a future release of Oracle Solaris. Scripts, programs, or procedures that use this command will likely need modification when upgrading to future Solaris software releases.The command line options provided with the `in.dhcpd`

daemon are used only for the current session, and include only some of the server options you can set. The dhcpsvc.conf(4) contains all the server default settings, and can be modified by using the dhcpmgr utility. See dhcpsvc.conf(4) for more details.

オプション

The following options are supported:

-b automatic | manual

This option enables BOOTP compatibility mode, allowing the DHCP server to respond to BOOTP clients. The option argument specifies whether the DHCP server should automatically allocate permanent lease IP addresses to requesting BOOTP clients if the clients are not registered in the DHCP network tables (automatic) or respond only to BOOTP clients who have been manually registered in the DHCP network tables (manual). This option only affects DHCP server mode.

-d

Debugging mode. The daemon remains as a foreground process, and displays verbose messages as it processes DHCP and/or BOOTP datagrams. Messages are displayed on the current TTY. This option can be used in both DHCP/BOOTP server mode and BOOTP relay agent mode.

-h *relay_hops*

Specifies the maximum number of relay agent hops that can occur before the daemon drops the DHCP/BOOTP datagram. The default number of relay agent hops is 4. This option affects both DHCP/BOOTP server mode and BOOTP relay agent mode.

-i *interface, . . .*

Selects the network interfaces that the daemon should monitor for DHCP/BOOTP datagrams. The daemon ignores DHCP/BOOTP datagrams on network interfaces not specified in this list. This option is only useful on machines that have multiple network interfaces. If this option is not specified, then the daemon listens for DHCP/BOOTP datagrams on all network interfaces. The option argument consists of a comma-separated list of interface names. It affects both DHCP/BOOTP server and BOOTP relay agent run modes.

-l *syslog_local_facility*

The presence of this option turns on transaction logging for the DHCP server or BOOTP relay agent. The value specifies the syslog local facility (an integer from 0 to 7 inclusive) the DHCP daemon should use for tagging the transactions. Using a facility separate from the LOG_DAEMON facility allows the network administrator to capture these transactions separately from other DHCP daemon events for such

purposes as generating transaction reports. See syslog(3C), for details about local facilities. Transactions are logged using a record with 9 space-separated fields as follows:

1. Protocol:

```
Relay mode:     "BOOTP"
Server mode:    "BOOTP" or "DHCP" based upon client
                     type.
```

2. Type:

```
Relay mode:     "RELAY-CLNT", "RELAY-SRVR"
Server mode:    "ASSIGN", "EXTEND", "RELEASE",
                    "DECLINE", "INFORM", "NAK" "ICMP-ECHO."
```

3. Transaction time: absolute time in seconds (unix time)
4. Lease time:

```
Relay mode:     Always 0.
Server mode:    0 for ICMP-ECHO events, absolute time in
                    seconds (unix time)  otherwise
```

5. Source IP address: Dotted Internet form

   Relay mode: Relay interface IP on RELAY-CLNT, INADDR_ANY on RELAY-SRVR.
   Server mode: Client IP.

6. Destination IP address: Dotted Internet form

   Relay mode: Client IP on RELAY-CLNT, Server IP on RELAY-SRVR.
   Server mode: Server IP.

7. Client Identifier: Hex representation (0-9, A-F)

   Relay mode: MAC address
   Server mode: BOOTP - MAC address; DHCP - client id

8. Vendor Class identifier (white space converted to periods (.)).

   Relay mode: Always "N/A"
   Server mode: Vendor class ID tokenized by converting white space characters to periods (.)

9. MAC address: Hex representation (0-9, A-F)

Relay mode: MAC address
Server mode: MAC address

The format of this record is subject to change between releases.

Transactions are logged to the console if daemon is in debug mode (`-d`).

Logging transactions impact daemon performance.

It is suggested that you periodically rotate the DHCP transaction log file to keep it from growing until it fills the filesystem. This can be done in a fashion similar to that used for the general system message log `/var/adm/messages` and is best accomplished using the facilities provided by `logadm(1M)`.

`-n`

Disable automatic duplicate IP address detection. When this option is specified, the DHCP server does not attempt to verify that an IP address it is about to offer a client is not in use. By default, the DHCP server pings an IP address before offering it to a DHCP/BOOTP client, to verify that the address is not in use by another machine.

`-o` *DHCP_offer_time*

Specifies the number of seconds the DHCP server should cache the offers it has extended to discovering DHCP clients. The default setting is `10` seconds. On slow network media, this value can be increased to compensate for slow network performance. This option affects only DHCP server mode.

`-r` *IP_address* | *hostname*, . . .

This option enables BOOTP relay agent mode. The option argument specifies a comma-separated list of IP addresses or hostnames of DHCP or BOOTP servers to which the relay agent is to forward BOOTP requests. When the daemon is started in this mode, any DHCP tables are ignored, and the daemon simply acts as a BOOTP relay agent.

A BOOTP relay agent listens to UDP port 68, and forwards BOOTP request packets received on this port to the destinations specified on the command line. It supports the `BROADCAST` flag described in RFC 1542. A BOOTP relay

agent can run on any machine that has knowledge of local routers, and thus does not have to be an Internet gateway machine.

Note that the proper entries must be made to the `netmasks` database so that the DHCP server being served by the BOOTP relay agents can identify the subnet mask of the foreign BOOTP/DHCP client's network. See `netmasks(4)` for the format and use of this database.

`-t` *dhcptab_rescan_interval*

Specifies the interval in minutes that the DHCP server should use to schedule the automatic rereading of the `dhcptab` information. Typically, you would use this option if the changes to the `dhcptab` are relatively frequent. Once the contents of the `dhcptab` have stabilized, you can turn off this option to avoid needless reinitialization of the server.

`-v`

Verbose mode. The daemon displays more messages than in the default mode. Note that verbose mode can reduce daemon efficiency due to the time taken to display messages. Messages are displayed to the current TTY if the debugging option is used; otherwise, messages are logged to the `syslogd` facility. This option can be used in both DHCP/BOOTP server mode and BOOTP relay agent mode.

使用例

例 1 Starting a DHCP Server in BOOTP Compatibility Mode

The following command starts a DHCP server in BOOTP compatibility mode, permitting the server to automatically allocate permanent IP addresses to BOOTP clients which are not registered in the server's table; limits the server's attention to incoming datagrams on network devices `le2` and `tr0`; drops BOOTP packets whose hop count exceeds 2; configures the DHCP server to cache extended DHCP offers for 15 seconds; and schedules `dhcptab` rescans to occur every 10 minutes:

```
# in.dhcpd -i le2,tr0 -h 2 -o 15 -t 10 -b automatic
```

例 2 Starting the Daemon in BOOTP Relay Agent Mode

The following command starts the daemon in BOOTP relay agent mode, registering the hosts `bladerunner` and `10.0.0.5` as relay destinations, with debugging and verbose modes enabled, and drops BOOTP packets whose hop count exceeds 5:

```
# in.dhcpd -d -v -h 5 -r bladerunner,10.0.0.5
```

ファイル

- `/etc/inet/dhcpsvc.conf`
- `/etc/init/hosts`
- `/usr/lib/inet/dhcp/nsu/rfc2136.so.1`

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dhcp
Interface Stability	Obsolete

関連項目

svcs(1), cron(1M), dhcpmgr(1M), dhtadm(1M), inetadm(1M), inetd(1M), logadm(1M), pntadm(1M), svcadm(1M), syslogd(1M), syslog(3C), dhcpsvc.conf(4), dhcp_network(4), dhcptab(4), ethers(4), hosts(4), netmasks(4), nsswitch.conf(4), attributes(5), dhcp(5), smf(5)

『Oracle Solaris の管理: IP サービス』

Alexander, S., and R. Droms, DHCP Options and BOOTP Vendor Extensions, RFC 2132, Silicon Graphics, Inc., Bucknell University, March 1997.

Droms, R., Interoperation Between DHCP and BOOTP, RFC 1534, Bucknell University, October 1993.

Droms, R., Dynamic Host Configuration Protocol, RFC 2131, Bucknell University, March 1997.

Wimer, W., Clarifications and Extensions for the Bootstrap Protocol, RFC 1542, Carnegie Mellon University, October 1993.

注意事項

The in.dhcpd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/dhcp-server
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

in.discardd – UDP or TCP discard protocol service

形式

`in.discardd`

`FMRI` *svc*`:/internet/discard:default`

機能説明

FMRI stands for Fault Management Resource Identifier. It is used to identify resources managed by the Fault Manager. See `fmd(1M)` and `smf(5)`.

The `in.discardd` service provides the server-side of the discard protocol. This protocol is used for debugging and bandwidth measurement and is available on both TCP and UDP transports through port 9.

The `in.discardd` service is an `inetd(1M)` `smf(5)` delegated service. The `in.discardd` detects which transport is requested by examining the socket it is passed by the `inetd` daemon.

The discard service simply throws away any data it receives from the client.

TCP-based service
: Once a connection is established, the `in.discardd` discards any data received. No response is generated. The connection remains open until the client terminates it.

UDP-based service
: The `in.discardd` listens for UDP datagrams. When a datagram is received, the server discards it. No response is sent.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/legacy-network-services
Interface Stability	Committed

関連項目

`inetd(1M)`, `attributes(5)`, `smf(5)`

RFC 863

名前

in.echod – UDP or TCP echo protocol service daemon

形式

`in.echod`

`FMRI` *svc*`:/internet/echo:default`

機能説明

FMRI stands for Fault Management Resource Identifier. It is used to identify resources managed by the Fault Manager. See `fmd(1M)` and `smf(5)`.

The `in.echod` service provides the server-side of the echo protocol. This protocol is used for debugging and bandwidth measurement and is available on both TCP and UDP transports, through port 7.

The `in.echod` service is an `inetd(1M)` `smf(5)` delegated service. The `in.echod` detects which transport is requested by examining the socket it is passed by the `inetd` daemon.

TCP-based service
: Once a connection is established, the `in.echod` echoes any data received from the client back to the client. The server echoes data until the client program terminates the connection.

UDP-based service
: The `in.echod` listens for UDP datagrams. When a datagram is received, the server creates a UDP datagram containing the data it received and sends it to the client.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/legacy-network-services
Interface Stability	Committed

関連項目

`inetd(1M)`, `attributes(5)`, `smf(5)`

RFC 862

名前

inetadm – inetd が制御するサービスを監視または構成する

形式

```
inetadm
inetadm -?
inetadm -p
inetadm -l {FMRI | pattern}
inetadm -e {FMRI | pattern}
inetadm -d {FMRI | pattern}
inetadm -m {FMRI | pattern}... {name=value}...
inetadm -M {name=value}...
```

機能説明

inetadm ユーティリティーは、inetd が管理する SMF サービスに次の機能を提供します。

- インストールされているすべてのサービスの一覧を表示します。
- サービスのプロパティーと値の一覧を表示します。
- サービスを有効に、または無効にすることができます。
- サービスのプロパティー値を変更したり、inetd によって提供されるデフォルト値を変更したりできます。

SMF サービスについては、smf(5) を参照してください。

引数を指定しない場合、inetadm は inetd(1M) によって制御されるすべてのサービス (現在の実行状態や有効であるかどうかなどの属性を含む) を表示します。

オプション

オプションが 1 つ以上の FMRI オペランド (FMRI の説明については、smf(5) を参照) をとるときに、オペランドが (サービスインスタンスではなく) サービスを指定し、そのサービスが単一のインスタンスしか持たない場合、inetadm はそのインスタンス上で動作します。

サービス名が指定されていてサービス名に複数のインスタンスが含まれている場合、またはパターンが指定されていてパターンが複数のインスタンスに一致する場合は、警告メッセージが表示され、そのオペランドは無視されます。

name=value パラメータをとるオプションに使用できる名前と許可されている値の説明は、inetd(1M) のマニュアルページに記載されています。

次のオプションがサポートされています。

-?	使用方法に関するメッセージを表示します。
-p	inetd によって提供されるすべてのデフォルトの inet サービスプロパティー値を、*name=value* の組の形式で一覧表示しま

す。値がブール型の場合は、TRUE または FALSE で表示されます。

-l {*FMRI* | *pattern*}...	指定されたサービスインスタンスのすべてのプロパティーを、*name=value* の組で一覧表示します。また、inetd によって提供されたデフォルト値からプロパティー値が継承されている場合、その *name=value* の組には「default」というトークンも表示されます。プロパティーの継承は、プロパティーにサービスインスタンスのデフォルトが指定されていない場合に発生します。
-e {*FMRI* | *pattern*}...	指定されたサービスインスタンスを有効にします。
-d {*FMRI* | *pattern*}...	指定されたサービスインスタンスを無効にします。
-m {*FMRI* | *pattern*}...{*name=value*}...	識別されたサービスインスタンスの指定されたプロパティーの値を変更します。プロパティーは、*name=value* の組をスペースで区切って指定します。インスタンス固有の値を削除し、プロパティーのデフォルト値を使うように指定するには、たとえば、name= のように、値なしでプロパティーを指定します。
-M {*name=value*}...	指定された inetd のデフォルトのプロパティー値を変更します。プロパティーは、*name=value* の組をスペースで区切って指定します。

使用例

例 1 サービスのプロパティーの表示

次のコマンドは spray サービスのプロパティーを表示します。

```
# inetadm -l network/rpc/spray:default
SCOPE    NAME=VALUE
         name="sprayd"
         endpoint_type="tli"
         proto="datagram_v"
         isrpc=TRUE
         rpc_low_version=1
         rpc_high_version=1
         wait=TRUE
         exec="/usr/lib/netsvc/spray/rpc.sprayd"
         user="root"
default  bind_addr=""
```

例 1　サービスのプロパティーの表示　　*(続き)*

```
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
default  connection_backlog=10
```

例 2　デフォルトプロパティーの表示

次のコマンドは、デフォルトプロパティーを表示します。

```
# inetadm -p
NAME=VALUE
bind_addr=""
bind_fail_max=-1
bind_fail_interval=-1
max_con_rate=-1
max_copies=-1
con_rate_offline=-1
failrate_cnt=40
failrate_interval=60
inherit_env=TRUE
tcp_trace=FALSE
tcp_wrappers=FALSE
default  connection_backlog=10
```

例 3　サービスのプロパティー値の変更

次のコマンドは、spray サービスの rpc_high_version を 3 に、tcp_trace を TRUE に変更します。

```
# inetadm -m network/rpc/spray:default \
         rpc_high_version=3 tcp_trace=TRUE
# inetadm -l network/rpc/spray:default
SCOPE    NAME=VALUE
         name="sprayd"
         endpoint_type="tli"
         proto="datagram_v"
         isrpc=TRUE
         rpc_low_version=1
         rpc_high_version=3
         wait=TRUE
```

例 3 サービスのプロパティー値の変更 (続き)

```
         exec="/usr/lib/netsvc/spray/rpc.sprayd"
         user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
         tcp_trace=TRUE
default  tcp_wrappers=FALSE
default  connection_backlog=10
```

終了ステータス

次の終了値が返されます。

0 操作は正常に完了しました。

1 致命的なエラーが発生しました。詳細情報は同時に表示されるエラーメッセージに記述されます。

2 無効な引数 (あいまいなサービス FMRI またはパターンなど) が指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

inetd(1M), svccfg(1M), attributes(5), smf(5)

名前

inetconv – inetd.conf エントリを smf サービスマニフェストに変換し、それらを smf リポジトリ内にインポートする

形式

```
inetconv -?

inetconv [-f] [-n] [-i srcfile] [-o destdir]

inetconv -e [-n] [-i srcfile]
```

機能説明

inetconv ユーティリティーは、inetd.conf(4) のレコードを含むファイルを smf(5) サービスマニフェストに変換したあと、それらのマニフェストを smf リポジトリ内にインポートします。inetd.conf ファイルの変換処理がいったん完了すると、inetadm(1M) ユーティリティーを使用しないと各 inet サービスの特性を変更できなくなります。

入力ファイル内の 1 つのサービス行と、それに対して生成されるマニフェストとの間には、1 対 1 の対応関係が成り立ちます。マニフェストはデフォルトで、次のテンプレートに従って命名されます。

<*svcname*>-<*proto*>.xml

<*svcname*> トークンはサービスの名前で、<*proto*> トークンはサービスのプロトコルで、それぞれ置き換えられます。ソース行のサービス名やプロトコルに含まれるスラッシュ (/) 文字はすべて、下線 (_) で置き換えられます。

各サービス行は、変換先となるサービスのプロパティーとして記録されます。

変換処理中に、不正なサービス行や inetd に対する内部的なサービス行が検出された場合には、マニフェストは生成されず、そのサービス行はスキップされます。

入力ファイルは変換処理の影響を受けず、元の状態に保たれます。

オプション

次のオプションがサポートされています。

- -? 使用方法に関するメッセージを表示します。
- -e 入力ファイル内に記述されている smf サービスを有効にします。
- -f このオプションが指定された場合、生成予定のサービスマニフェストと同じ名前のマニフェストが出力ディレクトリ内で見つかった場合に、inetconv はそのマニフェストを上書きします。それ以外の場合、エラーメッセージが生成され、そのサービスの変換は実行されません。
- -i *srcfile* 別の入力ファイル *srcfile* を指定できるようにします。このオプションが指定されなかった場合、inetd.conf(4) ファイルが入力として使用されます。
- -n 変換処理中に生成されたマニフェストの自動インポートを無効にします。その後、生成済みマニフェストを smf(5) リポジトリ内にインポートする必要が生じた場合、svccfg(1M) ユーティリティーを使えばそれを行うことができます。

-e オプションが指定された場合、-n オプションは単に、有効化される smf サービスを表示します。

-o　生成されたマニフェストに対する別の出力ディレクトリ *destdir* を指定できるようにします。このオプションが指定されなかった場合、RPC サービスのマニフェストは /lib/svc/manifest/network/rpc 内に、それ以外のサービスのマニフェストは /lib/svc/manifest/network 内に、それぞれ格納されます。

使用例

例 1　inetd.conf からの smf マニフェストの生成

次のコマンドは、inetd.conf(4) から smf(5) マニフェストを生成し、それらを /var/tmp 内に格納します。その際、同名の既存マニフェストはすべて上書きします。続いて、それらのマニフェストを smf リポジトリ内にインポートします。

```
# inetconv -f -o /var/tmp
100232/10 -> /var/tmp/100232_10-rpc_udp.xml
Importing 100232_10-rpc_udp.xml ...Done
telnet -> /var/tmp/telnet-tcp6.xml
Importing telnet-tcp6.xml ...Done
```

例 2　別の入力ファイルからのマニフェストの生成

次のコマンドは、異なる入力ファイルを指定しています。また、結果のマニフェストを smf リポジトリ内にロードしません。

```
# inetconv -n -i /export/test/inet.svcs -o /var/tmp
100232/10 -> /var/tmp/100232_10-rpc_udp.xml
telnet -> /var/tmp/telnet-tcp6.xml
```

終了ステータス

次の終了値が返されます。

0　処理が正常終了しました (エラーなし)。

1　無効なオプションが指定されました。

2　不正なサービス行が 1 つ以上存在します。それらに対するマニフェストは生成されませんでした。

3　1 つ以上の生成済みマニフェストのインポート中にエラーが発生しました。

4　システムエラーが発生しました。

ファイル

/lib/svc/manifest/network/{rpc}/<*svcname*>-<*proto*>.xml
　デフォルトの出力マニフェストファイル名。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

inetadm(1M), inetd(1M), svccfg(1M), inetd.conf(4), attributes(5), smf(5)

名前

inetd – Solaris サービス管理機能の inet サービス委任リスタータ

形式

```
inetd [configuration-file] start | stop | refresh

svc:/network/inetd:default
```

機能説明

inetd は Solaris サービス管理機能 (SMF) のインターネットサービスの委任リスタータです。その基本的な役割は、管理要求、システム障害、およびサービス障害に対応してサービスの状態を管理することと、必要に応じてサービスのネットワーク要求を待機することです。

inetd 構成ファイル inetd.conf(4) の編集によってサービスを管理することはなくなりました。代わりに、inetconv(1M) を使用して構成ファイルの内容を SMF 形式のサービスに変換してから、inetadm(1M) および svcadm(1M) を使用してこれらのサービスを管理します。inetconv でサービスを変換したあとは、inetd 構成ファイルのレガシーデータの変更は何の効果も持たなくなります。ただし、構成ファイルの変更を検出すると、inetd は管理者に警告します。詳細については、「inetd のメソッド」の start に関する説明を参照してください。

現在の inetd は SMF の外部からは実行できないことにも留意してください。つまり、以前の inetd でサポートされていたようなコマンド行からの実行はできません。これを試みると、標準エラー出力にメッセージが出力され、以前の inetd と SMF バージョンの inetd でサポートされているオプションのマッピングが表示されます。

inetd は、online 状態または degraded 状態になっているすべてのサービスに代わって接続を待機します。ユーザーがサービスを有効にすると、サービスはこれらの状態のどちらかになり、inetd はそのサービスに代わって待機を試みます。すでに別のサーバー (スタンドアロンまたはサードパーティーのインターネットサービス) が同じポートを待機している場合は、待機の試みが失敗することがあります。このような場合、inetd はこの状況をログに記録し、設定されている間隔で、設定されている回数だけ、ポートへのバインドを試行し続けます。詳細については、後述の「サービスのプロパティー」の bind_fail_max プロパティーを参照してください。

inetd で管理されるすべての SMF サービスの構成は、inetd の起動時に読み取られます。inetd が SMF 要求に対応して、あるいは SIGHUP シグナルを受信したために更新されるときに、再度読み取られます。構成の更新動作については、「inetd のメソッド」の refresh に関する説明を参照してください。

inetadm(1M) または svccfg(1M) ユーティリティーを使用すると、SMF リポジトリ内のインターネットサービスの構成を変更できます。inetadm には、svccfg と比較して、インターネット/RPC サービスコンテキストを提供するという利点があります。

サービスの状態

inetd は、サービス管理の役割の一部として、管理対象サービスごとに状態マシンを実装します。このマシンの状態は、smf(5) の状態セットから成ります。これらの状態のセマンティクスは次のとおりです。

uninitialized
このサービスはまだ inetd で処理されていません。

online
サービスは新しいネットワーク要求を処理しており、既存の接続がアクティブになっている可能性があります。

degraded
サービスがこの状態になっているのは、サービスに指定されているプロトコルについて要求の待機と処理を所定の回数だけ再試行しても、プロトコルの一部にしか成功しなかったためです。既存のネットワーク接続がアクティブになっている可能性があります。

offline
接続がアクティブになっている可能性はありますが、新しい要求は処理されていません。これは一時的な状態です。サービスは次のいずれかの理由で offline になっている可能性があります。

- サービスの依存関係がまだ処理されていません。依存関係が処理されると、サービスの状態は再評価されます。
- サービスは、設定されている接続レートの制限 max_con_rate を超えました。サービスの状態は、接続オフラインタイマー con_rate_offline がタイムアウトしたときに再評価されます。
- サービスは、アクティブな接続の許容数 max_copies に到達しました。サービスの状態は、アクティブな接続の数が max_copies より少なくなったときに再評価されます。
- inetd は、サービスのすべてのプロトコルについて、サービスに代わって待機することに失敗しました。前述のとおり、inetd は、設定されている間隔で、設定されている最大回数まで再試行します。サービスの状態は、待機が成功するか、再試行の制限回数に達したときに再評価されます。

disabled
サービスは管理者によって無効化されているため、新しい接続は受け入れておらず、アクティブな接続もありません。この状態を終了するには管理者の操作が必要です。

maintenance
サービスがこの状態になっているのは、動作異常のため管理者の対応を必要としているか、管理者がこの状態を要求したためです。

動作異常となるイベントには次のようなものがあります。サービスのプロトコルのいずれかについて、inetd がサービスに代わって待機できないまま、サービスのバインド再試行の制限回数を超えた。start 以外のメソッドから成功以外の戻り値が返されました。サービスがその障害レートを超えた。

サービスにパッチの適用などの管理を行う場合には、保守状態を要求します。この状態では、新しい要求は処理されませんが、既存の接続がアクティブになっている可能性はあります。この状態を終了するには管理者の操作が必要です。

管理対象サービスの現在の状態を取得するには、inetadm(1M) を使用します。

サービスのメソッド

inetd は、状態遷移の一環として、サービスで提供されている一連のメソッドのうち、設定されているものがあればそれを実行します。次の一連のメソッドがサポートされています。

inetd_start
online 状態または degraded 状態のサービスに対する要求を処理するために実行されます。アクティブな接続のあるサービスを区別する状態はないため、このメソッドは状態遷移の一環としては実行されません。

inetd_offline
サービスが online 状態または degraded 状態から offline 状態になるときに実行されます。このメソッドの実行により、その時点で独自の待機を実行している wait タイプのサービスは、待機を終了します。online 状態または degraded 状態のサービスが無効化される場合、このメソッドは disable メソッドの前に実行されます。wait タイプのサービスにはこのメソッドを実装する必要があります。

inetd_online
サービスが offline 状態から online 状態に遷移するときに実行されます。サービスの作成者は、このメソッドを使用すると、サービスで要求の処理を開始する前に何らかの準備を行うことができます。

inetd_disable
サービスが offline 状態から disabled 状態に遷移するときに実行されます。このメソッドの実行により、サービスにアクティブな接続がある場合はすべて終了されます。

inetd_refresh
次の両方の条件が満たされる場合に実行されます。

- inetd がフレームワークまたは SIGHUP によって更新される、または、サービスを更新する要求が届く
- サービスは現在 online 状態であり、サービスをいったん offline にしてから元に戻す必要が生じるような構成変更はない。

必須のメソッドは inetd_start だけです。ほかのメソッドが指定されていない場合、inetd はメソッドを実行しませんが、1 つのメソッドを正常に実行したかのように振る舞います。

サービスのプロパティー

SMF によって管理されるサービスの構成は SMF リポジトリに保存されます。構成は、サービスの基本構成、各サービスのメソッドの構成、および inetd によって管理されるすべてのサービスに適用されるデフォルトの構成から成ります。

サービスの構成とデフォルト値の表示および変更について詳しくは、inetadm(1M) を参照してください。

サービスの基本構成は、サービスの inetd というプロパティーグループに保存されます。基本構成は次のプロパティーから成ります。

bind_addr
: サービスのバインド先となるネットワークインタフェースのアドレス。空の文字列値を指定すると、サービスは任意のネットワークインタフェースで接続を受け入れます。

bind_fail_interval
: バインド失敗から次の再試行までの時間間隔 (秒単位)。値 0 と -1 を指定すると、再試行は行われず、最初の失敗で bind_fail_max を超えたかのように処理されます。

bind_fail_max
: inetd がサービスに関連付けられたポートへのバインドを試みる最大回数。値 -1 を指定すると、再試行回数は制限されません。設定されている制限に到達したときに、サービスのどのプロトコルもバインドされていない場合、サービスは maintenance 状態になります。あるいは、プロトコルの一部しかバインドされていない場合、サービスは degraded 状態になります。

con_rate_offline
: 設定されている最大接続レート max_con_rate を超えた場合にサービスを offline 状態に保つ時間 (秒単位)。値 0 および -1 を指定すると、接続レートの制限は無効になります。

connection_backlog
: バックログキューサイズ。着信するクライアント要求をサーバーの待機エンドポイントでキューに入れることのできる数の制限を表します。

endpoint_type
: サービスで使用されるソケットのタイプ。または、TLI ベースのサービスを表す値 tli。ソケットタイプの有効な値は、stream、dgram、raw、seqpacket です。

failrate_cnt
: サービスの障害レート制限の回数部分。障害レート制限は wait タイプのサービスに適用されます。一定時間内に *count* 個のサービスインスタンスが開始されると、この制限に到達します。このレートを超えると、サービスは maintenance 状態に遷移します。この動作は、10 分間隔で無期限に再試行が続けられていた以前の inetd の動作とは異なります。failrate_cnt 検査では、サービス要求を処理する前に障害が発生した、適切に動作していないサーバーが処理されます。障害レート制限がないと、このようなサーバーは再起動を繰り返してシステムリソースを大量に使用することになります。障害レートは、以前の inetd の -r オプションと同等です。値 0 および -1 を指定すると、この機能は無効になります。

`failrate_interval`
サービスの障害レートの時間部分 (秒単位)。値 `0` および `-1` を指定すると、障害レート制限機能は無効になります。

`inherit_env`
true の場合、`inetd` の環境をサービスの start メソッドに渡します。この設定にかかわらず、`inetd` は start メソッドの環境の変数 `SMF_FMRI`、`SMF_METHOD`、および `SMF_RESTARTER` を設定し、メソッドコンテキストに設定されている環境変数があればそれらも設定します。これらの変数については `smf_method(5)` の説明を参照してください。

`isrpc`
true の場合、これは RPC サービスです。

`max_con_rate`
`nowait` タイプのサービスに許容される最大接続レート (1 秒あたりの接続数)。値 `0` および `-1` を指定すると、接続レートの制限は無効になります。

`max_copies`
同時に実行できる `nowait` サービスの最大コピー数。値 `0` および `-1` を指定すると、コピー数の制限は無効になります。

`name`
次の値のいずれかに設定できます。

- `getservbyname(3SOCKET)` で認識されるサービス名。
- `isrpc` が `true` に設定されている場合は、`getrpcbyname(3NSL)` で認識されるサービス名。
- `isrpc` が `true` に設定されている場合は、有効な RPC プログラム番号。

`proto`
ソケットベースのサービスの場合は、サービスでサポートされているプロトコルのリスト。有効なプロトコルは、`tcp`、`tcp6`、`tcp6only`、`udp`、`udp6`、および `udp6only` です。TLI サービスの場合は、サービスでサポートされている `getnetconfigent(3NSL)` で認識される netid のリスト、および値 `tcp6only` と `udp6only`。RPC/TLI サービスの場合は、このリストに nettype も使用できます。`inetd` は最初に、リストのメンバーをこれらのサービスタイプの nettype として解釈しようとします。値 `tcp6only` と `udp6only` は、`inetd` に新しく導入されたもので、IPv4 のマップされた要求ではなく真の `IPv6` 要求だけを待機して渡すよう `inetd` に指示します。後述の「ソケットベースのサービスのプロトコルの構成」を参照してください。

`rpc_low_version`
サポートされるもっとも低い RPC バージョン。`isrpc` が `true` に設定されている場合は必須です。

`rpc_high_version`
サポートされるもっとも高い RPC バージョン。`isrpc` が `true` に設定されている場合は必須です。

tcp_keepalive
: true の場合、接続されているソケットでの定期的なメッセージ送信を有効にします。接続の相手側がこれらのメッセージに応答できない場合、接続は切断されているとみなされます。これが適用されるのは、endpoint_type が streams、かつ wait が false に設定されているサービスだけです。

tcp_trace
: true の場合、これが nowait タイプのサービスであれば、inetd は、syslog(3C) 機能を使用して、着信する接続ごとにクライアントの IP アドレスと TCP ポート番号、およびサービスの名前をログに記録します。inetd は syslog 機能の code デーモンと notice 優先度を使用します。syslog のコードと重要度については、syslog.conf(4) を参照してください。このログ機能は、TCP ラッパー機能によるログ機能とは独立しています。

  tcp_trace は、以前の inetd の -t オプション (および /etc/default/inetd のプロパティー ENABLE_CONNECTION_LOGGING) と同等です。

tcp_wrappers
: true の場合、TCP ラッパーによるアクセス制御を有効にします。これが適用されるのは、endpoint_type が streams、かつ wait が false に設定されているサービスだけです。syslog 機能の code デーモンを使用して、許可された接続 (notice 重要度を使用) および拒否されたトラフィック (warning 重要度を使用) をログに記録します。syslog のコードと重要度については、syslog.conf(4) を参照してください。TCP ラッパー機能とその構成ファイルのインタフェースの安定性は「流動的」です。TCP ラッパー機能は Sun で管理されているものではなく、リリース間の非互換性もまれではありません。attributes(5) を参照してください。

  TCP ラッパーの構成の詳細については、tcpd(1M) および hosts_access(4) のマニュアルページを参照してください。これらは、Solaris オペレーティングシステムの一部として /usr/sfw/man に用意されています。これらのページは、/usr/man にある標準の Solaris マニュアルページには属していません。

  tcp_wrappers は、以前の inetd の /etc/default/inetd のプロパティー ENABLE_TCPWRAPPERS と同等です。

wait
: true の場合、これは wait タイプのサービスです。それ以外の場合は nowait タイプのサービスです。wait タイプのサービスには次の特徴があります。

  - サービスの inetd_start メソッドは、実行されたときに、サービスのバインドされたエンドポイントで待機する役割を引き受けます。
  - inetd は、このメソッドが実行されたあと終了するのを待ってから、待機の役割を再開します。

  データグラムサーバーは、指定されたサービスにバインドされているサービスの提供にかかわる元のデータグラムエンドポイントで常に呼び出されるため、wait タイプとして構成する必要があります。これらの「待機」ソケットと「受け入

れ」ソケットは分離されていません。TCP ストリームサービスのように接続指向のサービスは、wait タイプと nowait タイプのどちらででも設計できます。

サービスの基本プロパティーのいくつかは省略可能です。省略されている場合、その値は inetd サービスの defaults プロパティーグループにあるデフォルト値から取り出されます。このようなプロパティーとそのシード値は次のとおりです。これらの値は inetadm(1M) で設定できます。

```
bind_fail_interval  -1
bind_fail_max       -1
con_rate_offline    -1
connection_backlog  10
failrate_count      40
failrate_time       60
inherit_env         true
max_con_rate        -1
max_copies          -1
tcp_keepalive       false
tcp_trace           false
tcp_wrappers        false
```

サービスに指定されている各メソッドの構成は、SMF リポジトリの、メソッドと同じ名前のプロパティーグループ内に保存されます。これらのメソッドに使用できるプロパティーセットには、svc.startd(1M) の管理対象サービスに指定されるプロパティーも含まれています。(詳細については、svc.startd(1M) を参照してください。)さらに、inetd_start メソッドには arg0 プロパティーを設定できます。

arg0 プロパティーを設定すると、inetd のサービスで外部ラッパープログラムを使用できるようになります。具体的には、サービスの start メソッドの第 1 引数 argv[0] に、サーバープログラムのパス以外のものを指定できるようになります。

外部ラッパープログラムを使用する場合に、サービスのデーモンに引数を渡すには、引数を exec プロパティーでラッパープログラムの引数として組み込むようにしてください。例:

```
exec='/path/to/wrapper/prog service_daemon_args'
arg0='/path/to/service/daemon'
```

smf_method(5) で説明されている特殊なメソッドトークンに加え、inetd では wait タイプのサービスに :kill_process トークンも使用できます。その結果は、:kill トークンを指定した場合と同じ動作になりますが、kill シグナルは wait タイプのサービスの start メソッドの親プロセスだけに送信され、そのプロセス契約の全メンバーには送信されない点が異なります (process(4) を参照)。

ソケットベースのサービスのプロトコルの構成

ソケットベースのサービスの inetd を構成する場合は、サービスでサポートしているものに応じて、前述の proto プロパティーの値を選択します。使用する proto 値のガイドラインを次に示します。

- IPv4 だけをサポートするサービスの場合: tcp と udp

- IPv6 だけをサポートするサービスの場合: tcp6only と udp6only
- IPv4 と IPv6 の両方をサポートするサービスの場合:
  - 廃止および非推奨: tcp6 と udp6
  - 推奨: proto フィールドだけが異なる 2 つのエントリを使用します。一方のエントリに tcp、もう一方のエントリに tcp6only を指定するか、udp と udp6only を指定します。

IPv4 と IPv6 の両方をサポートするサービスの構成例については、「使用例」を参照してください。

inetd のメソッド

inetd には、マスターリスタータ svc.startd(1M) で使用できる次のメソッドが用意されています。

start
: inetd によるサービスの提供を開始します。このメソッドにより、inetd は、その管理対象サービスの smf 要求と、online 状態または degraded 状態になっているサービスのネットワーク要求の処理を開始します。

  さらに、inetd は、監視している inetd.conf(4) 形式の構成ファイルが inetconv(1M) 変換の最後の実行以降に変更されているかどうかを確認します。変更されている場合、変更を適用するには inetconv を再度実行する必要があるという管理者へのメッセージが syslog に記録されます。

stop
: inetd によるサービスの提供を終了します。この時点で、inetd では、maintenance 状態または disabled 状態のサービスを除き、必要に応じてメソッドを実行して各サービスを offline 状態に遷移させます。

refresh
: 各管理対象サービスの更新を実行し、inetd.conf(4) 形式の構成ファイルが変更されているかどうかを確認します (start メソッドと同様)。サービスが更新されたときの動作は、その現在の状態によって異なります。

  - maintenance 状態または disabled 状態の場合は何も実行されません。構成は、サービスがこの状態でなくなったときに読み取られて処理されます。
  - offline 状態の場合は、構成が読み取られ、変更があればすぐに処理されます。
  - online 状態または degraded 状態の場合は、再バインドが必要になるような構成の変更があれば、サービスはいったん offline 状態になったあと、バインドの新しい構成を使用して元に戻ります。
  - online 状態の場合で再バインドが不要のときは、サービスの inetd_refresh メソッドが指定されていればそれが実行されます。これにより、online 状態の wait タイプのサービスはその他の変更を処理できます。

オプション

サポートされているオプションはありません。

オペランド

configuration-file
レガシーサービスファイル (inetd.conf(4)) に別の場所を指定します。

start|stop|refresh
実行する inetd のメソッドを指定します。

使用例

例 1 IPv4 と IPv6 の両方をサポートするサービスの構成

次のコマンドは、IPv4 と IPv6 の両方をサポートするサービスが存在することを確認し、それらに proto プロパティーを割り当てる方法を示しています。

```
example# svcs -a | grep mysvc
online         15:48:29 svc:/network/mysvc:dgram4
online         15:48:29 svc:/network/mysvc:dgram6
online         15:51:47 svc:/network/mysvc:stream4
online         15:52:10 svc:/network/mysvc:stream6

# inetadm -M network/rpc/mysvc:dgram4 proto=udp
# inetadm -M network/rpc/mysvc:dgram6 proto=udp6only
# inetadm -M network/rpc/mysvc:stream4 proto=tcp
# inetadm -M network/rpc/mysvc:stream6 proto=tcp6only
```

これらのコマンドの説明については、svcs(1) と inetadm(1M) を参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

fmd(1M), inetadm(1M), inetconv(1M), svcadm(1M), svccfg(1M), svcs(1), svc.startd(1M), syslog(3C), getnetconfigent(3NSL), getrpcbyname(3NSL), getservbyname(3SOCKET), inetd.conf(4), process(4), syslog.conf(4), attributes(5), smf(5), smf_method(5)

注意事項

inetd デーモンは、Solaris 9 リリース以前の Solaris オペレーティングシステムにある同じ名前のデーモンと比較して、実行する機能は同じですが、実装方法が大きく異なっています。Solaris の現在のリリースでは、inetd は Solaris サービス管理機能 (smf(5) を参照) の一部であり、この機能の内部でのみ実行されます。

/etc/default/inetd ファイルは非推奨になりました。ENABLE_CONNECTION_LOGGING プロパティーと ENABLE_TCP_WRAPPERS プロパティーで表されていた機能は、現在はそれぞれ tcp_trace プロパティーと tcp_wrappers プロパティーで使用可能です。これらのプロパティーについては、前述の「サービスのプロパティー」を参照してください。

名前

in.fingerd, fingerd – remote user information server

形式

```
/usr/sbin/in.fingerd [-s]
```

機能説明

fingerd implements the server side of the Name/Finger protocol, specified in RFC 742. The Name/Finger protocol provides a remote interface to programs which display information on system status and individual users. The protocol imposes little structure on the format of the exchange between client and server. The client provides a single command line to the finger server which returns a printable reply.

fingerd waits for connections on TCP port 79. Once connected, it reads a single command line terminated by RETURN-LINEFEED and passes the arguments to finger(1), prepended with -s. fingerd closes its connections as soon as the output is finished.

オプション

fingerd supports the following option.

-s　Enable secure mode. Deny forwarding of queries to other remote hosts.

ファイル

/var/adm/utmpx	User and accounting information.
/etc/passwd	System password file.
/var/adm/lastlog	Last login times.
$HOME/.plan	User's plans.
$HOME/.project	User's projects.

使用法

fingerd and in.fingerd are IPv6–enabled. See ip6(7P).

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

finger(1), svcs(1), inetadm(1M), inetd(1M), svcadm(1M), attributes(5), smf(5), ip6(7P)

Harrenstien, Ken, RFC 742, NAME/FINGER, Network Information Center, SRI International, Menlo Park, Calif., December 1977.

注意事項

Connecting directly to the server from a TIP or an equally narrow-minded TELNET-protocol user program can result in meaningless attempts at option negotiation being sent to the server, which foul up the command line interpretation. fingerd should be taught to filter out IAC's and perhaps even respond negatively (IAC does not) to all option commands received.

The in.fingerd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/finger:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

infocmp – compare or print out terminfo descriptions

形式

```
/usr/bin/infocmp [-d] [-c] [-n] [-I] [-L] [-C] [-r] [-u]
     [-s | d | i | l | c] [-v] [-V] [-1] [-w width]
     [-A directory] [-B directory] [termname]...
```

機能説明

`infocmp` compares a binary `terminfo` entry with other terminfo entries, rewrites a `terminfo` description to take advantage of the `use=` terminfo field, or prints out a `terminfo` description from the binary file (`term`) in a variety of formats. It displays boolean fields first, then numeric fields, followed by the string fields. If no options are specified and zero, or one *termname* is specified, the `-I` option is assumed. If more than one *termname* is specified, the `-d` option is assumed.

オプション

The `-d` , `-c` , and `-n` options can be used for comparisons. `infocmp` compares the `terminfo` description of the first terminal *termname* with each of the descriptions given by the entries for the other terminal's *termname*. If a capability is defined for only one of the terminals, the value returned will depend on the type of the capability: `F` for boolean variables, –`1` for integer variables, and `NULL` for string variables.

`-d` Produce a list of each capability that is different between two entries. This option is useful to show the difference between two entries, created by different people, for the same or similar terminals.

`-c` Produce a list of each capability that is common between two entries. Capabilities that are not set are ignored. This option can be used as a quick check to see if the `-u` option is worth using.

`-n` Produce a list of each capability that is in neither entry. If no *termname* is given, the environment variable `TERM` will be used for both of the *termname*s. This can be used as a quick check to see if anything was left out of a description.

The `-I` , `-L` , and `-C` options will produce a source listing for each terminal named.

`-I` Use the `terminfo` names.

`-L` Use the long C variable name listed in < `term.h` >.

`-C` Use the `termcap` names. The source produced by the `-C` option may be used directly as a `termcap` entry, but not all of the parameterized strings may be changed to the `termcap` format. `infocmp` will attempt to convert most of the parameterized information, but anything not converted will be plainly marked in the output and commented out. These should be edited by hand.

`-r` When using `-C` , put out all capabilities in `termcap` form.

If no *termname* is given, the environment variable `TERM` will be used for the terminal name.

All padding information for strings will be collected together and placed at the beginning of the string where `termcap` expects it. Mandatory padding (padding information with a trailing '/') will become optional.

All termcap variables no longer supported by terminfo , but are derivable from other terminfo variables, will be displayed. Not all terminfo capabilities will be translated; only those variables which were part of termcap will normally be displayed. Specifying the -r option will take off this restriction, allowing all capabilities to be displayed in termcap form.

Note that because padding is collected to the beginning of the capability, not all capabilities are displayed. Mandatory padding is not supported. Because termcap strings are not as flexible, it is not always possible to convert a terminfo string capability into an equivalent termcap format. A subsequent conversion of the termcap file back into terminfo format will not necessarily reproduce the original terminfo source.

Some common terminfo parameter sequences, their termcap equivalents, and some terminal types which commonly have such sequences, are:

```
terminfo     termcap      Representative Terminals
%p1%c    %.    adm
%p1%d    %d    hp, ANSI standard, vt100
%p1%'x'%+%c    %+x    concept
%i    %i    ANSI standard, vt100
%p1%?%'x'%>%t%p1%'y'%+%;    %>xy    concept
%p2 is printed before %p1    %r    hp
```

-u Produce a terminfo source description of the first terminal *termname* which is relative to the sum of the descriptions given by the entries for the other terminals' *termname*s. It does this by analyzing the differences between the first *termname* and the other *termnames* and producing a description with use= fields for the other terminals. In this manner, it is possible to retrofit generic terminfo entries into a terminal's description. Or, if two similar terminals exist, but were coded at different times, or by different people so that each description is a full description, using infocmp will show what can be done to change one description to be relative to the other.

A capability is displayed with an at-sign (@) if it no longer exists in the first *termname*, but one of the other *termname* entries contains a value for it. A capability's value is displayed if the value in the first *termname* is not found in any of the other *termname* entries, or if the first of the other *termname* entries that has this capability gives a different value for that capability.

The order of the other *termname* entries is significant. Since the terminfo compiler tic does a left-to-right scan of the capabilities, specifying two use= entries that contain differing entries for the same capabilities will produce different results, depending on the order in which the entries are given. infocmp will flag any such inconsistencies between the other *termname* entries as they are found.

Alternatively, specifying a capability *after* a use= entry that contains, it will cause the second specification to be ignored. Using infocmp to recreate a description can be a useful check to make sure that everything was specified correctly in the original source description.

Another error that does not cause incorrect compiled files, but will slow down the compilation time, is specifying superfluous `use=` fields. `infocmp` will flag any superfluous `use=` fields.

`-s` Sorts the fields within each type according to the argument below:

- `d` Leave fields in the order that they are stored in the `terminfo` database.
- `i` Sort by `terminfo` name.
- `l` Sort by the long C variable name.
- `c` Sort by the `termcap` name.

If the `-s` option is not given, the fields are sorted alphabetically by the `terminfo` name within each type, except in the case of the `-C` or the `-L` options, which cause the sorting to be done by the `termcap` name or the long C variable name, respectively.

`-v` Print out tracing information on standard error as the program runs.

`-V` Print out the version of the program in use on standard error and exit.

`–1` Print the fields one to a line. Otherwise, the fields are printed several to a line to a maximum width of 60 characters.

`-w`*width* Changes the output to *width* characters.

The location of the compiled `terminfo` database is taken from the environment variable `TERMINFO` . If the variable is not defined, or the terminal is not found in that location, the system `terminfo` database, usually in `/usr/share/lib/terminfo`, is used. The options `-A` and `-B` may be used to override this location.

`-A` *directory* Set `TERMINFO` for the first *termname*.

`-B` *directory* Set `TERMINFO` for the other *termname*s. With this, it is possible to compare descriptions for a terminal with the same name located in two different databases. This is useful for comparing descriptions for the same terminal created by different people.

ファイル

`/usr/share/lib/terminfo/?/*` Compiled terminal description database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`captoinfo(1M)`, `tic(1M)`, `curses(3CURSES)`, `terminfo(4)`, `attributes(5)`

名前

in.iked – daemon for the Internet Key Exchange (IKE)

形式

```
/usr/lib/inet/in.iked [-d] [-f filename] [-p level]
/usr/lib/inet/in.iked -c [-f filename]
```

機能説明

in.iked performs automated key management for IPsec using the Internet Key Exchange (IKE) protocol.

in.iked implements the following:

- IKE authentication with either pre-shared keys, DSS signatures, RSA signatures, or RSA encryption.
- Diffie-Hellman key derivation using either 768, 1024, 1536, 2048, 3072, or 4096-bit public key moduli, or 256, 384, or 521-bit elliptic curve moduli.
- Authentication protection with cipher choices of AES, DES, Blowfish, or 3DES, and hash choices of either HMAC-MD5 or HMAC-SHA-1. Encryption in in.iked is limited to the IKE authentication and key exchange. See ipsecesp(7P) for information regarding IPsec protection choices.

in.iked is managed by the following smf(5) service:

```
svc:/network/ipsec/ike
```

This service is delivered disabled because the configuration file needs to be created before the service can be enabled. See ike.config(4) for the format of this file.

See “Service Management Facility” for information on managing the smf(5) service.

in.iked listens for incoming IKE requests from the network and for requests for outbound traffic using the PF_KEY socket. See pf_key(7P).

in.iked has two support programs that are used for IKE administration and diagnosis: ikeadm(1M) and ikecert(1M).

The ikeadm(1M) command can read the /etc/inet/ike/config file as a rule, then pass the configuration information to the running in.iked daemon using a doors interface.

```
example# ikeadm read rule /etc/inet/ike/config
```

Refreshing the ike smf(5) service provided to manage the in.iked daemon sends a SIGHUP signal to the in.iked daemon, which will (re)read /etc/inet/ike/config and reload the certificate database.

The preceding two commands have the same effect, that is, to update the running IKE daemon with the latest configuration. See “Service Management Facility” for more details on managing the in.iked daemon.

When Trusted Extensions are enabled (see labeld(1M)), in.iked can be used in the global zone to negotiate labeled security associations. On labeled systems using in.iked, UDP ports 500 and 4500 must be configured as multi-level ports for the global zone in the tnzonecfg file (see tnzonecfg(4), part of the Solaris Trusted Extensions Reference Manual). See ike.config(4) for more information on configuring in.iked to be label-aware.

Service Management Facility

The IKE daemon (in.iked) is managed by the service management facility, smf(5). The following group of services manage the components of IPsec:

```
svc:/network/ipsec/ipsecalgs   (See ipsecalgs(1M))
svc:/network/ipsec/policy      (See ipsecconf(1M))
svc:/network/ipsec/manual-key  (See ipseckey(1M))
svc:/network/ipsec/ike         (see ike.config(4))
```

The manual-key and ike services are delivered disabled because the system administrator must create configuration files for each service, as described in the respective man pages listed above.

The correct administrative procedure is to create the configuration file for each service, then enable each service using svcadm(1M).

The ike service has a dependency on the ipsecalgs and policy services. These services should be enabled before the ike service. Failure to do so results in the ike service entering maintenance mode.

If the configuration needs to be changed, edit the configuration file then refresh the service, as follows:

```
example# svcadm refresh ike
```

The following properties are defined for the ike service:

config/admin_privilege
: Defines the level that ikeadm(1M) invocations can change or observe the running in.iked. The acceptable values for this property are the same as those for the -p option. See the description of -p in OPTIONS.

config/config_file
: Defines the configuration file to use. The default value is /etc/inet/ike/config. See ike.config(4) for the format of this file. This property has the same effect as the -f flag. See the description of -f in OPTIONS.

config/debug_level
: Defines the amount of debug output that is written to the debug_logfile file, described below. The default value for this is op or operator. This property controls the recording of information on events such as re-reading the configuration file. Acceptable value for debug_level are listed in the ikeadm(1M) man page. The value all is equivalent to the -d flag. See the description of -d in OPTIONS.

`config/debug_logfile`

Defines where debug output should be written. The messages written here are from debug code within `in.iked`. Startup error messages are recorded by the `smf(5)` framework and recorded in a service-specific log file. Use any of the following commands to examine the `logfile` property:

```
example# svcs -l ike
example# svcprop ike
example# svccfg -s ike listprop
```

The values for these log file properties might be different, in which case both files should be inspected for errors.

`config/ignore_errors`

A boolean value that controls `in.iked`'s behavior should the configuration file have syntax errors. The default value is `false`, which causes `in.iked` to enter maintenance mode if the configuration is invalid.

Setting this value to `true` causes the IKE service to stay online, but correct operation requires the administrator to configure the running daemon with `ikeadm(1M)`. This option is provided for compatibility with previous releases.

These properties can be modified using `svccfg(1M)` by users who have been assigned the following authorization:

```
solaris.smf.value.ipsec
```

PKCS#11 token objects can be unlocked or locked by using `ikeadm` token login and `ikeadm` token logout, respectively. Availability of private keying material stored on these PKCS#11 token objects can be observed with: `ikeadm dump certcache`. The following authorizations allow users to log into and out of PKCS#11 token objects:

```
solaris.network.ipsec.ike.token.login
solaris.network.ipsec.ike.token.logout
```

See `auths(1)`, `ikeadm(1M)`, `user_attr(4)`, `rbac(5)`.

The service needs to be refreshed using `svcadm(1M)` before a new property value is effective. General, non-modifiable properties can be viewed with the `svcprop(1)` command.

```
# svccfg -s ipsec/ike setprop config/config_file = \
/new/config_file
# svcadm refresh ike
```

Administrative actions on this service, such as enabling, disabling, refreshing, and requesting restart can be performed using `svcadm(1M)`. A user who has been assigned the authorization shown below can perform these actions:

```
solaris.smf.manage.ipsec
```

The service's status can be queried using the `svcs(1)` command.

The `in.iked` daemon is designed to be run under `smf(5)` management. While the `in.iked` command can be run from the command line, this is discouraged. If the `in.iked` command is to be run from the command line, the `ike` `smf(5)` service should be disabled first. See `svcadm(1M)`.

オプション

The following options are supported:

`-c` Check the syntax of a configuration file.

`-d` Use debug mode. The process stays attached to the controlling terminal and produces large amounts of debugging output. This option is deprecated. See "Service Management Facility" for more details.

`-f` *filename* Use *filename* instead of `/etc/inet/ike/config`. See `ike.config(4)` for the format of this file. This option is deprecated. See "Service Management Facility" for more details.

`-p` *level* Specify privilege level (*level*). This option sets how much `ikeadm(1M)` invocations can change or observe about the running `in.iked`.

Valid *levels* are:

0 Base level

1 Access to preshared key info

2 Access to keying material

If `-p` is not specified, *level* defaults to `0`.

This option is deprecated. See "Service Management Facility" for more details.

セキュリティー

This program has sensitive private keying information in its image. Care should be taken with any core dumps or system dumps of a running `in.iked` daemon, as these files contain sensitive keying information. Use the `coreadm(1M)` command to limit any corefiles produced by `in.iked`.

ファイル

`/etc/inet/ike/config`	Default configuration file.
`/etc/inet/secret/ike.privatekeys/*`	Private keys. A private key *must* have a matching public-key certificate with the same filename in `/etc/inet/ike/publickeys/`.
`/etc/inet/ike/publickeys/*`	Public-key certificates. The names are only important with regard to matching private key names.
`/etc/inet/ike/crls/*`	Public key certificate revocation lists.
`/etc/inet/secret/ike.preshared`	IKE pre-shared secrets for Phase I authentication.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

svcs(1), coreadm(1M), ikeadm(1M), ikecert(1M), labeld(1M), svccfg(1M), svcadm(1M), ike.config(4), attributes(5), smf(5), ipsecesp(7P), pf_key(7P)

See tnzonecfg(4), part of the Solaris Trusted Extensions Reference Manual.

Harkins, Dan and Carrel, Dave. RFC 2409, Internet Key Exchange (IKE). Network Working Group. November 1998.

Maughan, Douglas, Schertler, M., Schneider, M., Turner, J. RFC 2408, Internet Security Association and Key Management Protocol (ISAKMP). Network Working Group. November 1998.

Piper, Derrell, RFC 2407, The Internet IP Security Domain of Interpretation for ISAKMP. Network Working Group. November 1998.

Fu, D.; Solinos, J., RFC 4753, ECP Groups for IKE and IKEv2. Network Working Group. January 2007.

Lepinski, M.; Kent, S., RFC 5114, Additional Diffie-Hellman Groups for Use with IETF Standards. Network Working Group. January 2008.

名前

init – プロセス制御の初期化

形式

```
/usr/sbin/init [0123456abcQqSs]
```

機能説明

init はデフォルトの原始ユーザープロセスです。(ブート中にカーネルに渡されるオプションによっては、別の原始ユーザープロセスが呼び出されることもあります。kernel(1M)を参照してください)。init はサービス管理機能の中心のコンポーネントである、svc.configd(1M) および svc.startd(1M) を初期化し、初期化に失敗した場合はこれらのコンポーネントを再開始します。下位互換性を保つため、init は /etc/inittab に従って汎用プロセスの起動および再起動も行います (下記参照)。

下記の実行レベルおよびシステムブートの説明は、互換性を保つ目的でのみ記述されており、それ以外の場合はサービス管理機能 smf(5) によって廃止されています。

init の障害

init がシステムのシャットダウン以外の何らかの理由で終了した場合、プロセス ID 1 で再起動されます。

定義されている実行レベル

任意の時点で、システムは 8 つの実行レベルのいずれか 1 つにあります。実行レベルはソフトウェア構成で、そこでは選択されたプロセスグループだけが存在します。各実行レベルに対する、init によって生成されるプロセスは、/etc/inittab で定義されています。init は 8 つの実行レベル、0–6 および S または s (S と s は同じ) のいずれか 1 つを取ることができます。実行レベルは、特権ユーザーが /usr/sbin/init を実行すると変更されます。

init とシステムのブート

システムがブートされると、init が呼び出され、次のイベントが発生します。最初に、svc:/system/environment:init サービスのプロパティーを読み取ります。これらのプロパティーには、LANG や LC_CTYPE などのロケール関連の環境に使用する値があります。次に init は、/etc/inittab で initdefault エントリを探します (inittab(4) を参照)。initdefault のエントリが、

存在する場合
: オプションまたはマイルストーンのプロパティーが svc.startd(1M) に指定されていない場合のみ、init は通常、このエントリで指定された実行レベルを開始時の最初の実行レベルとして使用します。

存在しない場合
: サービス管理機能 smf(5) は、svc.startd(1M) で指定された構成を調べ、オプションまたはマイルストーンのプロパティーで指定されたマイルストーンを入力します。

/etc/inittab 内の initdefault エントリは、次の実行レベルに対応します。

S または s
: init はシングルユーザー状態に移行します。この状態では、システムコンソールデバイス (/dev/console) が読み取り/書き込み用に開かれ、コマンド /usr/sbin/su (su(1M) を参照) が呼び出されます。init を使用して、システムの

実行レベルを変更します。(ファイルの終わり (EOF) を使用して) シェルを終了させたが、/etc/inittab が存在しない場合、init は再びシングルユーザー状態に戻すだけです。

0-6
init は対応する実行レベルに移行します。実行レベル 0、5、および 6 はシステム停止用に予約されています。実行レベル 2、3、および 4 は、マルチユーザー操作用に使用できます。

電源投入後に、init が最初のシングルユーザー状態以外の実行レベルに移行する場合、init はまず、/etc/inittab を走査して boot および bootwait エントリを探します (inittab(4) を参照)。移行する実行レベルがエントリの実行レベルと一致している場合は、/etc/inittab の他のプロセスが実行される前に、これらのエントリが実行されます。このようにすると、ファイルシステムのマウントなど、オペレーティングシステムの特殊な初期化を先に終えてから、ユーザーにシステムを使用させることができます。次に init は、/etc/inittab を走査し、その実行レベルで処理すべき他のすべてのエントリを実行します。

/etc/inittab の各プロセスを生成するために、init は各エントリを読み込み、再生成が必要なエントリごとに、子プロセスを生成します。/etc/inittab で指定された全プロセスを生成すると、init は、子孫プロセスの消滅、powerfail シグナル、またはシステムの実行レベルを変更する他の init プロセスからのシグナルのいずれかを待機します。これらの状態のいずれか 1 つが発生すると、init は /etc/inittab を再検査します。

inittab に関するその他の事項

/etc/inittab にはいつでも新しいエントリを追加できますが、init は前述の 3 つの状態のいずれかが発生するまで待機し続けるので、/etc/inittab を再検証しません。この状況を回避するには、init Q または init q コマンドを使用して、init が /etc/inittab をただちに再検査するようにします。

ブート時に init が呼び出されたとき、また、システムがシングルユーザー状態から別の実行状態に移行するたびに、init はコンソールの ioctl(2) 状態をファイル /etc/ioctl.syscon に格納されているモードに設定します。init はシングルユーザー状態が移行するたびに、このファイルに書き込みを行います。

実行レベルの変更

実行レベルの変更が要求されると、init は目標の実行レベルで定義されていないすべてのプロセスに警告シグナル (SIGTERM) を送ります。init は 5 秒間待機したのち、終了シグナル (SIGKILL) を送信することによって、これらのプロセスを強制終了させます。さらに、init は、実行レベルが変更中であることを svc.startd(1M) に通知します。すると、svc.startd(1M) は、その実行レベル変更に対応するマイルストーンが依存している一連のサービスに、システムを制限します。

init によって生成されたプロセスが終了したことを通知するシグナルを受信すると、init は、その事実と原因を /var/adm/utmpx と /var/adm/wtmpx (存在していれば) に記録します (who(1) を参照)。生成されたプロセスの履歴は /var/adm/wtmpx に記録されます。

init は powerfail シグナル (SIGPWR) を受信した場合、/etc/inittab を走査し、タイプが powerfail と powerwait の特殊なエントリを探します。これらのエントリが呼び出されてから (実行レベルが許可する場合)、その後の処理が実行されます。このように、init はオペレーティングシステムの停止中にさまざまなクリーンアップおよび記録機能を実行します。

環境変数の設定

タイムゾーンや文字の書式設定といった環境変数のデフォルト値を svc:/system/environment:init サービスのプロパティーリストで設定できます。

セキュリティー

init は pam(3PAM) を使用してセッションを管理します。/etc/pam.conf に記述されている PAM 構成ポリシーにより、init に使用されるセッション管理モジュールが指定されます。次に、pam.conf ファイルの抜粋を示します。UNIX セッション管理モジュールを使用するinit のエントリが指定されています。

```
init    session    required    pam_unix_session.so.1
```

init サービスに対応するエントリがない場合、「other」のサービスのエントリが使用されます。

オプション

0
ファームウェアモードに移行します。

1
システムをシステム管理者モードにします。すべてのローカルファイルシステムがマウントされます。いくつかの重要なコアプロセスだけが実行を続けます。このモードはオプションのユーティリティーパッケージをインストールする場合など、管理作業を行うためのものです。すべてのファイルにアクセスできます。ユーザーはシステムにログインできません。

この要求は、smf(5) がシステムのマイルストンを次の値に制限するための要求に対応しています。svc: /milestone/single-user: default。

2
システムをマルチユーザーモードにします。すべてのマルチユーザー環境用の端末プロセスおよびデーモンが生成されます。一般に、この状態を、マルチユーザー状態といいます。

この要求は、smf(5) がシステムのマイルストンを次の値に制限するための要求に対応しています。svc: /milestone/multi-user: default。

3
ネットワークを介してローカルリソースを使用できるようにすることで、マルチユーザーモードを拡張します。

この要求は、smf(5) がシステムのマイルストンを次の値に制限するための要求に対応しています。svc: /milestone/multi-user-server: default。

4
代替のマルチユーザー環境構成として定義できます。システム運用上は必要なく、通常は使用されません。

5
電源を切っても安全なようにマシンを停止します。可能であれば電源を自動的に切断します。

6
オペレーティングシステムを停止したあと、/etc/inittab の initdefault エントリに定義されている状態でリブートします。

サービス svc:/system/boot-config:default はデフォルトで有効になっています。config/fastreboot_default プロパティーが true に設定されている場合に init 6 を指定すると、システムの特定の機能に応じて、特定のファームウェア初期化手順およびテスト手順が省略されます。

a、b、c
/etc/inittab エントリに a、b、または c という実行レベルが設定されているときだけ処理します。これらは擬似状態であり、特定のコマンドを実行するように定義できますが、現在の実行レベルが変更されるわけではありません。

Q、q
/etc/inittab を再検査します。

S、s
シングルユーザーモードに移行します。適切な形式の /etc/inittab ファイルがなくてもかまわない唯一の実行レベルです。このファイルが存在しない場合、デフォルトで init が入ることができる唯一の正当な実行レベルはシングルユーザーモードだけです。シングルユーザーモードでは、基本システム動作に必要なファイルシステムがマウントされます。システムがシングルユーザーモードに移行すると、これらのファイルシステムは (リモートファイルサーバーが提供している場合でも) マウントされたままになります。他のローカルファイルシステムもマウントされたままになります。シングルユーザーモードへの切り替え時に、init または init.d によって開始された、マルチユーザーモードでしか実行してはならないプロセスはすべて強制終了されます。さらに、utmpx エントリが設定されているプロセスもすべて強制終了されます。この最後の状態では、SAC が起動したすべてのポートモニターが確実に強制終了され、これらのポートモニターによって起動されたサービスも、ttymon login ログインサービスを含めてすべて強制終了されます。

この要求は、smf(5) がシステムのマイルストンを次の値に制限するための要求に対応しています。svc: /milestone/single-user: default。

ファイル

/dev/console
システムコンソールデバイス。

/etc/default/init
このファイルは廃止されたため、将来のリリースで削除される可能性があります。init プロセスではこのファイルから値を取得するのではなく svc:/system/environment:init サービスのプロパティーを読み取ります。以前に /etc/default/init の編集によって行われた変更を反映させるために、システム

管理者またはシステム構成権利プロファイルを持つ管理者は init サービスインスタンスの対応するプロパティーを設定し、そのインスタンスを更新できます。

この読み取り専用ファイルには、環境変数とそのデフォルト値が含まれています。変数は次のとおりです。

TZ
常に localtime に設定されます。システムのタイムゾーンを設定するためには、管理者は SMF サービス timezone:default の timezone/localtime プロパティーを設定する必要があります。

CMASK
init が使用し、すべてのプロセスが init プロセスから継承するマスク (umask(1) を参照)。設定されていない場合、init はカーネルから継承したマスクを使用します。CMASK の設定に関係なく、init は必ず、022 で umask の適用を試みてからファイルを作成します。

LC_CTYPE
文字の種類の情報

LC_MESSAGES
翻訳メッセージ

LC_MONETARY
通貨の書式情報

LC_NUMERIC
数値の書式情報

LC_TIME
時刻の書式情報

LC_ALL
設定されている場合、他のすべての LC_* 環境変数でこの値が使用されます。

LANG
LC_ALL が設定されてなく、かつ特定の LC_* も設定されていない場合は、これら環境変数に LANG の値が使用されます。

/etc/inittab
init によるプロセスディスパッチ制御。

/etc/ioctl.syscon
シングルユーザー状態に移行したときに init によって保存された、コンソールの ioctl (入出力制御) 状態。

/etc/svc/volatile/init.state
障害から回復するのに必要な init 状態

/var/adm/utmpx
ユーザーアクセスおよび管理情報。

/var/adm/wtmpx
ユーザーアクセスと管理情報の履歴

/var/run/initpipe
内部通信用の名前付きパイプ。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

login(1), sh(1), stty(1), who(1), kernel(1M), shutdown(1M), su(1M), svc.configd(1M), svc.startd(1M), ttymon(1M), ioctl(2), kill(2), ctime(3C), pam(3PAM), init.d(4), inittab(4), pam.conf(4), TIMEZONE(4), utmpx(4), attributes(5), pam_unix_session(5), smf(5), termio(7I)

診断

2 分間に 10 回以上、/etc/inittab 内の 1 つのエントリが再起動されていることを検出すると、init はそのエントリのコマンド文字列に誤りがあるとみなして、システムコンソール上にエラーメッセージを表示します。さらに 5 分経過するか、あるいはユーザーが生成した init コマンドからシグナルを受信するまで、init はそのエントリを再起動することを拒否します。このようにすることで、inittab ファイルに入力ミスがあった場合や /etc/inittabで参照されているプログラムが削除された場合でも、init がシステムリソースを使い果たすのを防止できます。

注意事項

init を実行できるのは、特権ユーザーだけです。

/etc/inittab で、S または s 状態をむやみに使用してはいけません。このファイルを変更するときに注意すべきことは、この状態を initdefault 以外の行に追加しないことです。

/etc/inittab の initdefault エントリでデフォルトの状態が指定されていない場合は、状態 6 になります。その結果、システムはファームウェアへのアクセスとリブートを繰り返すループに陥ります。

システムのブート時に utmpx ファイルを作成できない場合、システムは /etc/inittab の initdefault エントリで指定されている状態に関係なく、“s” の状態でブートします。/var ファイルシステムにアクセスできない場合には、この状況が発生することがあります。

システムが S または s 状態へ移行するときに、/etc/nologin ファイル (nologin(4) を参照) が作成されます。その後、実行レベル 2 に移行すると、/etc/rc2.d ディレクトリにあるスクリプトによってこのファイルが削除されます。

init は内部通信に名前付きパイプ /var/run/initpipe を使用します。

名前

init.sma – start and stop the snmpd daemon

形式

```
/etc/init.d/init.sma start | stop | restart | status
```

機能説明

The `init.sma` utility is run automatically during installation and each time the system is rebooted. This utility manages the `snmpd`.

オプション

The following options are supported:

`start` Starts the `snmpd` daemon.

`stop` Stops the `snmpd` daemon.

`restart` Stops then starts the `snmpd` daemon.

`status` Reports the `snmpd` daemon's status.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsmagt
Interface Stability	Uncommitted

関連項目

`attributes(5)`

名前

init.wbem – start and stop the CIM Boot Manager

形式

```
svccfg svc:/application/management/wbem start | stop
```

機能説明

The `init.wbem` utility is run automatically during installation, each time the system is rebooted, and by means of the `svccfg(1M)` command. This method manipulates the CIM Object Manager (CIMOM). `init.wbem` can be used to start, stop, or retrieve status from the server.

CIM Object Manager

The CIM Object Manager manages CIM objects on a WBEM-enabled system. A CIM object is a computer representation, or model, of a managed resource, such as a printer, disk drive, or CPU. CIM objects are stored internally as Java classes.

When a WBEM client application accesses information about a CIM object, the CIM Object Manager contacts either the appropriate provider for that object or the CIM Object Manager Repository. Providers are classes that communicate with managed objects to access data.

When a WBEM client application requests data from a managed resource that is not available from the CIM Object Manager Repository, the CIM Object Manager forwards the request to the provider for that managed resource. The provider dynamically retrieves the information.

At startup, the CIM Object Manager performs the following functions:

- Listens for RMI connections on RMI port 5987 and for XML/HTTP connections on HTTP port 5988.
- Sets up a connection to the CIM Object Manager Repository.
- Waits for incoming requests.

During normal operations, the CIM Object Manager performs the following functions:

- Performs security checks to authenticate user login and authorization to access namespaces.
- Performs syntactical and semantic checking of CIM data operations to ensure that they comply with the latest CIM Specification.
- Routes requests to the appropriate provider or to the CIM Object Manager Repository.
- Delivers data from providers and from the CIM Object Manager Repository to WBEM client applications.

A WBEM client application contacts the CIM Object Manager to establish a connection when it needs to perform WBEM operations, such as creating a CIM class or updating a CIM instance. When a WBEM client application connects to a CIM Object Manager, it gets a reference to the CIM Object Manager, which it then uses to request services and operations.

System Booting

The `init.wbem` script is installed in the `/etc/init.d` directory.

オプション

The `init.wbem` function is implemented as a service management facility (SMF) method and is controlled by means of the `svcadm(1M)` command using the fault management resource identifier (FMRI) `svc:/application/management/wbem`.

The following options are supported through `svcadm(1M)`:

`start` Starts the CIMOM on the local host.

`stop` Stops the CIMOM.

`status` Gets the status of the CIMOM.

The SMF property `options/tcp_listen` is used to control whether CIMOM and the Solaris Management console server will respond to requests from remote systems.

The specification:

```
svc:/application/management/wbem/options/tcp_listen = true
```

...allows remote access and `false` disallows remote access. `false` is the default value.

使用例

例 1 Allowing Access to Remote Systems

The following commands enable CIMOM to allow access from remote systems.

```
# svccfg -s \
svc:/application/management/wbem setprop options/tcp_listen = true
# svcadm refresh svc:/application/management/wbem
```

注意事項

When the `init.wbem` script is invoked by SMF, it does not run the CIMOM directly. The server process is in Java and is too heavyweight to be run immediately at system boot time. Instead, three lightweight processes listen on three different ports that the CIMOM normally uses. This acts similarly to `inetd(1M)`.

Because Java programs cannot inherit file descriptors as other programs can, there is a small time period from when the first connection is made until the server is fully operational where client connections may be dropped. WBEM clients are immune to this, as they will retry until the server comes online.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

`inetd(1M)`, `mofcomp(1M)`, `svccfg(1M)`, `svcadm(1M)`, `wbemadmin(1M)`, `wbemlogviewer(1M)`, `attributes(5)`, `smf(5)`

名前

inityp2l – create NIS (YP) to LDAP configuration files

形式

```
/usr/lib/netsvc/yp/inityp2l [-m mapping_file_name]
      [-c config_file_name]
```

機能説明

The `inityp2l` utility assists with creation of the `NISLDAPmapping` and `ypserv` files. See `NISLDAPmapping(4)` and`ypserv(4)`. `inityp2l` examines the NIS maps on a system. and through a dialogue with the user, determines which NIS to (and from) LDAP mappings are required. A `NISLDAPmapping` file is then created based on this information. The utility asks users about their LDAP server configuration and a `ypserv` file is created based on this information.

The `inityp2l` utility handles mappings for standard NIS maps and the `auto.*` series of maps. If requested, it creates default mappings for custom maps, with each map entry represented as a single DIT string. `inityp2l` does not handle full custom mapping, but if requested, `inityp2l` will insert comments into the `NISLDAPmapping` file that indicate where these should be added.

To write to the `NISLDAPmapping` or `ypserv` files is potentially dangerous. `inityp2l` warns the user and asks for confirmation before:

1. it overwrites either file
2. it writes to the default `NISLDAPmapping` file location, if this file did not previously exist. This is important because the existence of a file in this location causes NIS components to work NIS to LDAP (N2L) mode when next restarted, rather than to traditional NIS mode.

`inityp2l` assists with rapid creation of a simple N2L configuration files. It is not a general purpose tool for the management of these files. An advanced user who would like to maintain the files or use custom mappings should examine the output of `inityp2l` and customize it by using a standard text editor.

オプション

`inityp2l` supports the following options:

`-c` Specify the name of the generated `ypserv` file. The default location is described in ファイル.

`-m` Specify the name of the generated `NISLDAPmapping` file. The default is described in ファイル.

ファイル

`/var/yp`	The directory to be searched for candidate domains (`/var/yp/*`) and NIS maps (`/var/yp/*/*`)
`/var/yp/NISLDAPmapping`	The default location for the generated `NISLDAPmapping` file
`/etc/default/ypserv`	The default location for the generated `ypserv` file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis
Interface Stability	Obsolete

関連項目

NISLDAPmapping(4), ypserv(4), attributes(5)

名前

in.lpd – BSD print protocol adaptor

形式

`/usr/lib/print/in.lpd`

機能説明

`in.lpd` implements the network listening service for the BSD print protocol specified in RFC 1179. The BSD print protocol provides a remote interface for systems to interact with a local spooling system. The protocol defines five standard requests from the client to the server: starting queue processing, transferring print jobs, retrieving terse status, retrieving verbose status, and canceling print jobs.

The `in.lpd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/lp
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

`in.lpd` uses the `config/log_from_remote` property to allow or disallow remote access. The default value of this property, `localhost`, disallows remote access.

`inetd` waits for connections on TCP port 515. Upon receipt of a connect request, `in.lpd` is started to service the connection. Once the request has been filled, `in.lpd` closes the connection and exits.

使用例

例 1　Allowing Remote Access

The following command allows remote access to `in.lpd`.

```
# inetadm -m svc:/application/print/rfc1179:default bind_addr=""
```

終了ステータス

The following exit values are returned:

`0`	Successful completion.
non-zero	An error occurred.

ファイル

`/usr/lib/print/bsd-adaptor/bsd_*.so*`	Spooler translation modules.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/lp/print-client-commands

関連項目 svcs(1), inetadm(1M), inetd(1M), svcadm(1M), attributes(5), smf(5)

名前

in.mpathd – IP multipathing daemon

形式

/usr/lib/inet/in.mpathd

機能説明

The in.mpathd daemon performs failure and repair detection for IP interfaces that have been placed into an IPMP group (or optionally, for all IP interfaces on the system). It also controls which IP interfaces in an IPMP group are "active" (being used by the system to send or receive IP data traffic) in a manner that is consistent with the administrator's configured policy. Functioning interfaces that are not active are referred to as *alternate* interfaces.

The in.mpathd daemon can detect IP interface failure and repair through three methods: by monitoring the IFF_RUNNING flag for each IP interface (link-state based failure detection), by sending and receiving ICMP probes on each IP interface (ICMP probe-based failure detection), or by transitive probing. Failure detection by transitive probing is done by sending and receiving ICMP probes on active interfaces, while determining the liveness of alternate interfaces through their ability to send and receive link-layer (that is, transitive) probes to an active interface.

By default, only link-state based failure detection is enabled. This requires the driver to support link-status notification. ICMP probe-based failure detection must be enabled through the configuration of one or more test addresses, which are described below. Transitive probing can be enabled by modifying the value of the SMF property (shown below) to true:

```
svc:/network/ipmp/config/transitive-probing
```

See EXAMPLES for more information on how to modify the value of the transitive-probing property.

Both ICMP and transitive probe-based failure detection methods test the entire IP interface send and receive path. The ipmpstat(1M) utility can be used to check which failure detection methods are enabled.

If transitive-probing is set to true, and no test addresses are configured for a given IPMP group, then transitive probing will be used. If it is set to false (default value), then transitive probing will not be used under any circumstance.

If only link-state based failure detection is enabled, then the health of the interface is determined solely from the state of the IFF_RUNNING flag. If probes have been enabled, the interface is considered failed if either link-state or probes indicate a failure, and repaired once the failure detection method has indicated the failure has been corrected. Although all interfaces in a group need not be configured with the same failure detection methods, transitive probing will be disabled on any interface of a group that has at least one IP test address.

As mentioned above, to perform ICMP probe-based failure detection, in.mpathd requires a test address on each IP interface for the purpose of sending and receiving probes. Each address must be marked NOFAILOVER (see ifconfig(1M)) and in.mpathd will be limited to probing targets on the same subnet. Each address may be configured statically or acquired by means of

DHCP. To find targets, in.mpathd first consults the routing table for routes on the same subnet, and uses the specified next-hop. If no routes match, it sends all-hosts ICMP probes and selects a subset of the systems that respond. Thus, for probe-based failure detection to operate, there must be at least one neighbor on each subnet that responds to ICMP echo request probes. The ipmpstat(1M) utility can be used to display both the current probe target information and the status of sent probes.

Both IPv4 and IPv6 are supported. If an IP interface is plumbed for IPv4 and an IPv4 test address is configured then in.mpathd will start sending ICMPv4 probes over that IP interface. Similarly, if an IP interface is plumbed for IPv6 and an IPv6 test address is configured, then in.mpathd will start sending ICMPv6 probes over that IP interface. However, note that in.mpathd will ignore IPv6 test addresses that are not link-local. If both IPv4 and IPv6 are plumbed, it is sufficient to configure only one of the two, that is, either an IPv4 test address or an IPv6 test address. If both IPv4 and IPv6 test addresses are configured, in.mpathd probes using both ICMPv4 and ICMPv6.

As mentioned above, in.mpathd also controls which IP interfaces in an IPMP group are "active" (used by the system to send and receive IP data traffic). Specifically, in.mpathd tracks the administrative configuration of each IPMP group and attempts to keep the number of active IP interfaces in each group consistent with that configuration. Therefore, if an active IP interface fails, in.mpathd will activate an INACTIVE interface in the group, provided one exists (it will prefer INACTIVE interfaces that are also marked STANDBY). Likewise, if an IP interface repairs and the resulting repair leaves the IPMP group with more active interfaces than the administrative configuration specifies, in.mpathd will deactivate one of the interfaces (preferably one marked STANDBY), except when the FAILBACK variable is used, as described below. Similar adjustments will be made by in.mpathd when offlining IP interfaces (for instance, in response to if_mpadm(1M)).

The in.mpathd daemon accesses three variable values in /etc/default/mpathd: FAILURE_DETECTION_TIME, FAILBACK and TRACK_INTERFACES_ONLY_WITH_GROUPS.

The FAILURE_DETECTION_TIME variable specifies the probe-based failure detection time. The shorter the failure detection time, the more probe traffic. The default value of FAILURE_DETECTION_TIME is 10 seconds. This means that IP interface failure will be detected by in.mpathd within 10 seconds. The IP interface repair detection time is always twice the value of FAILURE_DETECTION_TIME. Note that failures and repairs detected by link-based failure detection are acted on immediately, though in.mpathd may ignore link state changes if it suspects that the link state is flapping due to defective hardware; see DIAGNOSTICS.

By default, in.mpathd limits failure and repair detection to IP interfaces that are configured as part of a named IPMP group. Setting TRACK_INTERFACES_ONLY_WITH_GROUPS to no enables failure and repair detection on all IP interfaces, even if they are not part of a named IPMP group. IP interfaces that are tracked but not part of a named IPMP group are considered to be part of the "anonymous" IPMP group. In addition to having no name, this IPMP group is special in that its IP interfaces are not equivalent and thus cannot take over for one another in

the event of an IP interface failure. That is, the anonymous IPMP group can only be used for failure and repair detection, and provides no high-availability or load-spreading.

As described above, when `in.mpathd` detects that an IP interface has repaired, it activates it so that it will again be used to send and receive IP data traffic. However, if `FAILBACK` is set to `no`, then the IP interface will only be activated if no other active IP interfaces in the group remain. However, the interface may subsequently be activated if another IP interface in the group fails.

SMF Management

The `in.mpathd` daemon service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/ipmp:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`.

使用例

例 1　Enabling Fault Detection by Transitive Probing

The following example shows the sequence of SMF commands used to enable fault detection by transitive probing.

```
# svccfg -s svc:/network/ipmp setprop config/transitive-probing=true
# svcadm refresh  svc:/network/ipmp:default
```

ファイル

`/etc/default/mpathd`　Contains default values used by the `in.mpathd` daemon.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`if_mpadm(1M)`, `ifconfig(1M)`, `ipmpstat(1M)`, `svcadm(1M)`, `attributes(5)`, `smf(5)`, `icmp(7P)`, `icmp6(7P)`,

『Oracle Solaris の管理: IP サービス』

診断

`IP interface` *interface_name* `has a hardware address which is not unique in group` *group_name*`; offlining`

説明: For probe-based failure detection, load-spreading, and other code IPMP features to work properly, each IP interface in an IPMP group must have a unique hardware address. If this requirement is not met, `in.mpathd` will automatically offline all but one of the IP interfaces with duplicate hardware addresses.

`IP interface` *interface_name* `now has a unique hardware address in group` *group_name*`; onlining`

説明: The previously-detected duplicate hardware address is now unique, and therefore `in.mpathd` has brought *interface_name* back online.

`Test address` *address* `is not unique in group; disabling probe-based failure detection on` *interface_name*

説明: For `in.mpathd` to perform probe-based failure detection, each test address in the group must be unique.

`No test address configured on interface` *interface_name* `disabling probe-based failure detection on it`

説明: For `in.mpathd` to perform probe-based failure detection on an IP interface, it must be configured with a test address: IPv4, IPv6, or both.

`IP` *interface_name* `in group` *group_name* `is not plumbed for IPv[4|6], affecting IPv[4|6] connectivity`

説明: All IP interfaces in a multipathing group must be homogeneously plumbed. For example, if one IP interface is plumbed for IPv4, then all IP interfaces in the group must be plumbed for IPv4, or IPv4 packets will not be able to be reliably sent and received. The STREAMS modules pushed on all IP interfaces must also be identical.

`The link has come up on` *interface_name* `more than 2 times in the last minute; disabling repair until it stabilizes.`

説明: To limit the impact of interfaces with intermittent hardware (such as a bad cable), `in.mpathd` will not consider an IP interface with a frequently changing link state as repaired until the link state stabilizes.

`Invalid failure detection time of` *time*`, assuming default 10000 ms`

説明: An invalid value was encountered for `FAILURE_DETECTION_TIME` in the `/etc/default/mpathd` file.

`Too small failure detection time of` *time*`, assuming minimum of 100 ms`

説明: The minimum value that can be specified for `FAILURE_DETECTION_TIME` is currently 100 milliseconds.

`Invalid value for FAILBACK` *value*

説明: Valid values for the boolean variable `FAILBACK` are `yes` or `no`.

`Invalid value for TRACK_INTERFACES_ONLY_WITH_GROUPS` *value*

説明: Valid values for the boolean variable `TRACK_INTERFACES_ONLY_WITH_GROUPS` are `yes` or `no`.

`Cannot meet requested failure detection time of` *time* `ms on (inet[6]` *interface_name*`) new failure detection time for group` *group_name* `is` *time* `ms`

説明: The round trip time for ICMP probes is higher than necessary to maintain the current failure detection time. The network is probably congested or the probe targets are loaded. `in.mpathd` automatically increases the failure detection time to whatever it can achieve under these conditions.

`Improved failure detection time` *time* `ms on (inet[6]` *interface_name*`) for group` *group_name*

説明: The round trip time for ICMP probes has now decreased and `in.mpathd` has lowered the failure detection time correspondingly.

`IP interface failure detected on` *interface_name*

説明: `in.mpathd` has detected a failure on *interface_name*, and has set the `IFF_FAILED` flag on *interface_name*, ensuring that it will not be used for IP data traffic.

`IP interface repair detected on` *interface_name*

説明: `in.mpathd` has detected a repair on *interface_name*, and has cleared the `IFF_FAILED` flag. Depending on the administrative configuration, the *interface_name* may again be used for IP data traffic.

`All IP interfaces in group` *group* `are now unusable`

説明: `in.mpathd` has determined that none of the IP interfaces in *group* can be used for IP data traffic, breaking network connectivity for the group.

`At least 1 IP interface (`*interface_name*`) in group` *group* `is now usable`

説明: `in.mpathd` has determined that at least one of the IP interfaces in *group* can again be used for IP data traffic, restoring network connectivity for the group.

`The link has gone down on` *interface_name*

説明: `in.mpathd` has detected that the `IFF_RUNNING` flag for *interface_name* has been cleared, indicating that the link has gone down.

`The link has come up on` *interface_name*

説明: `in.mpathd` has detected that the `IFF_RUNNING` flag for *interface_name* has been set, indicating that the link has come up.

名前

in.ndpd – daemon for IPv6 autoconfiguration

形式

`/usr/lib/inet/in.ndpd [-adt] [-f` *config_file*`]`

機能説明

`in.ndpd` provides both the host and router autoconfiguration components of Neighbor Discovery for IPv6 and Stateless and Stateful Address Autoconfiguration for IPv6. In particular, `in.ndpd` implements:

- router discovery;
- prefix discovery;
- parameter discovery;
- invocation of stateful address autoconfiguration;
- stateless address autoconfiguration; and
- privacy extensions for stateless address autoconfiguration.

Other aspects of Neighbor Discovery are implemented by `ip6(7P)`, including:

- address resolution;
- neighbor unreachability detection; and
- redirect.

The duplicate address detection function is implemented by the system kernel.

`in.ndpd` is managed by the service management facility (SMF), by means of the service identifier:

```
svc:/network/routing/ndp:default
```

If the `/etc/inet/ndpd.conf` file does not exist or does not set the variable `AdvSendAdvertisements` to true for a network interface, then `in.ndpd` will make the node a host for that interface, that is, sending router solicitation messages and then using router advertisement messages it receives to autoconfigure the node. Note that `in.ndpd` only autoconfigures the addresses of global or site-local scope from the prefix advertisement.

If `AdvSendAdvertisements` is set to true for an interface, then `in.ndpd` will perform router functions on that interface, that is, sending router advertisement messages to autoconfigure the attached hosts, but not use any advertisements it receives for autoconfiguration. However, when sending advertisements, `in.ndpd` will use the advertisements it sends itself to autoconfigure its prefixes.

Stateless autoconfiguration requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers. The stateless mechanism enables a host to generate its own addresses and uses local information as well as non-local information that is advertised by routers to generate the addresses. `in.ndpd` will plumb logical interfaces for each of these addresses.

Stateful autoconfiguration involves the `dhcpagent(1M)` daemon and the use of the DHCPv6 protocol. The `dhcpagent` daemon is responsible for plumbing the logical interfaces for the acquired addresses, maintaining the leases, and handling duplicate addresses. `in.ndpd` starts

the dhcpagent daemon automatically and signals when DHCPv6 should be started. in.ndpd also detects when dhcpagent configures the logical interfaces, and sets the appropriate prefix length on each according to received Routing Advertisement messages. in.ndpd will not stop dhcpagent; use ifconfig(1M) to control dhcpagent if necessary.

Temporary addresses that are autoconfigured for an interface can also be implemented. A temporary address token is enabled for one or more interfaces on a host. However, unlike standard, autoconfigured IPv6 addresses, a temporary address consists of the site prefix and a randomly generated 64 bit number. This random number becomes the interface ID segment of the IPv6 address. A link-local address is not generated with the temporary address as the interface ID.

If the kernel detects a duplicate temporary address, in.ndpd will automatically choose another.

Routers advertise all prefixes that have been assigned on the link. IPv6 hosts use Neighbor Discovery to obtain a subnet prefix from a local router. Hosts automatically create IPv6 addresses by combining the subnet prefix with an interface IDs that is generated from an interface's MAC address. In the absence of routers, a host can generate only link-local addresses. Link-local addresses can only be used for communication with nodes on the same link.

For information on how to enable IPv6 address autoconfiguration, see 『Oracle Solaris の管理: IP サービス』

オプション

Supported options and equivalent SMF service properties are listed below. SMF service properties are set using a command of the form:

```
# routeadm -m ndp:default key=value
```

-a
: Turn off stateless and stateful address auto configuration. When set, the daemon does not autoconfigure any addresses and does not renumber any addresses. This option does the same thing as the following lines in ndpd.conf(4):

```
ifdefault StatefulAddrConf off
ifdefault StatelessAddrConf off
```

Use of this option is equivalent to setting the stateless_addr_conf property to false.

-d
: Turn on large amounts of debugging output on stdout. When set, the program runs in the foreground and stays attached to the controlling terminal. Use of this option is equivalent to setting the debug property to true.

-f *config_file*
: Use *config_file* for configuration information instead of the default /etc/inet/ndpd.conf. Use of this option is equivalent to setting the config_file property to the configuration file to be used.

`-t`

Turn on tracing (printing) of all sent and received packets to `stdout`. When set, the program runs in the foreground and stays attached to the controlling terminal. As such, this option cannot be run under the SMF.

ファイル

`/etc/inet/ndpd.conf`	Configuration file. This file is not necessary on a host, but it is required on a router to enable `in.ndpd` to advertise autoconfiguration information to the hosts.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`dhcpagent(1M)`, `ifconfig(1M)`, `routeadm(1M)`, `svcadm(1M)`, `ndpd.conf(4)`, `attributes(5)`, `icmp6(7P)`,`ip6(7P)`, `attributes(5)`, `smf(5)`

『Oracle Solaris の管理: IP サービス』

Narten, T., Nordmark, E., Simpson, W.RFC 2461, Neighbor Discovery for IP Version 6 (IPv6). The Internet Society. December 1998.

Thomson, S., Narten, T. RFC 2462, IPv6 Stateless Address Autoconfiguration. The Internet Society. December 1998.

Narten, T., and Draves, R. RFC 3041, Privacy Extensions for Stateless Address Autoconfiguration in IPv6. The Internet Society. January 2001.

診断

Receipt of a `SIGHUP` signal will make `in.ndpd` restart and reread `/etc/inet/ndpd.conf`.

注意事項

The `in.ndpd` daemon service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/routing/ndp:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)` or `routeadm(1M)`.

名前

in.rarpd, rarpd – DARPA Reverse Address Resolution Protocol server

形式

```
/usr/sbin/in.rarpd [-d] -a
```

```
/usr/sbin/in.rarpd [-d] device unit
```

機能説明

`in.rarpd` starts a daemon that responds to Reverse Address Resolution Protocol (RARP) requests. The daemon forks a copy of itself that runs in background. It must be run as root.

RARP is used by machines at boot time to discover their Internet Protocol (IP) address. The booting machine provides its Ethernet address in a RARP request message. Using the `ethers` and `hosts` databases, `in.rarpd` maps this Ethernet address into the corresponding IP address which it returns to the booting machine in an RARP reply message. The booting machine must be listed in both databases for `in.rarpd` to locate its IP address. `in.rarpd` issues no reply when it fails to locate an IP address.

`in.rarpd` uses the STREAMS-based Data Link Provider Interface (DLPI) message set to communicate directly with the datalink device driver.

オプション

The following options are supported:

`-a` Get the list of available network interfaces from IP using the SIOCGIFADDR ioctl and start a RARP daemon process on each interface returned.

`-d` Print assorted debugging messages while executing.

使用例

例 1 Starting An in.rarpd Daemon For Each Network Interface Name Returned From /dev/ip:

The following command starts an `in.rarpd` for each network interface name returned from `/dev/ip`:

```
example# /usr/sbin/in.rarpd -a
```

例 2 Starting An in.rarpd Daemon On The Device /dev/le With The Device Instance Number 0

The following command starts one `in.rarpd` on the device `/dev/le` with the device instance number `0`.

```
example# /usr/sbin/in.rarpd le 0
```

ファイル

`/etc/ethers` File or other source, as specified by `nsswitch.conf`(4).

`/etc/hosts` File or other source, as specified by `nsswitch.conf`(4).

`/tftpboot`

`/dev/ip`

`/dev/arp`

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/network

関連項目

svcs(1), boot(1M), ifconfig(1M), svcadm(1M), ethers(4), hosts(4), netconfig(4), nsswitch.conf(4),attributes(5), smf(5), dlpi(7P)

Finlayson, R., Mann, T., Mogul, J., and Theimer, M., RFC 903, A Reverse Address Resolution Protocol, Network Information Center, SRI International, June 1984.

注意事項

The in.rarpd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rarp
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

in.rdisc, rdisc – network router discovery daemon

形式

```
/usr/sbin/in.rdisc  [-a] [-f] [-s] [send-address] [receive-address]
```

```
/usr/sbin/in.rdisc  -r [-p preference] [-T interval]
 [send-address] [receive-address]
```

機能説明

`in.rdisc` remains part of the software distribution of the Solaris Operating Environment. It is, however, not used by default. `in.routed(1M)` includes the functionality provided by `in.rdisc`. See `routeadm(1M)` for details of how to specify the IPV4 routing daemon.

`in.rdisc` implements the ICMP router discovery protocol. The first form of the command is used on hosts and the second form is used on routers.

`in.rdisc` can be invoked in either the first form (host mode) or second form (router mode).

On a host, `in.rdisc` populates the network routing tables with default routes. On a router, advertises the router to all the hosts.

`in.rdisc` is managed by the service management facility (SMF), by means of the service identifier:

```
svc:/network/routing/rdisc:default
```

Host (First Form)

On a host, `in.rdisc` listens on the `ALL_HOSTS` (224.0.0.1) multicast address for `ROUTER_ADVERTISE` messages from routers. The received messages are handled by first ignoring those listed router addresses with which the host does not share a network. Among the remaining addresses, the ones with the highest preference are selected as default routers and a default route is entered in the kernel routing table for each one of them.

Optionally, `in.rdisc` can avoid waiting for routers to announce themselves by sending out a few `ROUTER_SOLICITATION` messages to the `ALL_ROUTERS` (224.0.0.2) multicast address when it is started.

A timer is associated with each router address. The address will no longer be considered for inclusion in the routing tables if the timer expires before a new *advertise* message is received from the router. The address will also be excluded from consideration if the host receives an *advertise* message with the preference being maximally negative or with a lifetime of zero.

Router (Second Form)

When `in.rdisc` is started on a router, it uses the `SIOCGIFCONF` `ioctl(2)` to find the interfaces configured into the system and it starts listening on the `ALL_ROUTERS` multicast address on all the interfaces that support multicast. It sends out *advertise* messages to the `ALL_HOSTS` multicast address advertising all its `IP` addresses. A few initial *advertise* messages are sent out during the first 30 seconds and after that it will transmit *advertise* messages approximately every 600 seconds.

When `in.rdisc` receives a *solicitation* message, it sends an *advertise* message to the host that sent the *solicitation* message.

When `in.rdisc` is terminated by a signal, it sends out an *advertise* message with the preference being maximally negative.

オプション

Supported options and equivalent SMF service properties are listed below. SMF service properties are set using a command of the form:

```
# routeadm -m rdisc:default key=value
```

`-a` Accept all routers independent of the preference they have in their *advertise* messages. Normally, `in.rdisc` only accepts (and enters in the kernel routing tables) the router or routers with the highest preference. Use of this option is equivalent to setting the `accept_all` property to true.

`-f` Run `in.rdisc` forever even if no routers are found. Normally, `in.rdisc` gives up if it has not received any *advertise* message after soliciting three times, in which case it exits with a non-zero exit code. If `-f` is not specified in the first form then `-s` must be specified. For SMF execution, this option is required.

`-r` Act as a router, rather than a host. Use of this option is equivalent to setting the `act_as_router` property to true.

`-s` Send three *solicitation* messages initially to quickly discover the routers when the system is booted. When `-s` is specified, `in.rdisc` exits with a non-zero exit code if it can not find any routers. This can be overridden with the `-f` option. This option is not compatible with SMF execution and is not supported for the `rdisc` service.

`-p` *preference* Set the preference transmitted in the *solicitation* messages. The default is zero. Use of this option is equivalent to setting the `preference` property.

`-T` *interval* Set the interval between transmitting the *advertise* messages. The default time is 600 seconds. Use of this option is equivalent to setting the `transmit_interval` property.

The `send-address` and `receive-address` daemon options can be set by means of the `send_address` and `receive_address` properties.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/routing

関連項目

`in.routed(1M)`, `routeadm(1M)`, `svcadm(1M)`, `ioctl(2)`, `gateways(4)`, `attributes(5)`, `smf(5)`, `icmp(7P)`, `inet(7P)`

Deering, S.E., editor, ICMP Router Discovery Messages, RFC 1256, Network Information Center, SRI International, Menlo Park, California, September 1991.

名前

in.rexecd, rexecd – remote execution server

形式

```
in.rexecd
```

機能説明

`in.rexecd` is the server for the `rexec(3SOCKET)` routine. The server provides remote execution facilities with authentication based on user names and passwords. It is invoked automatically as needed by `inetd(1M)`, and then executes the following protocol:

1. The server reads characters from the socket up to a null (`\0`) byte. The resultant string is interpreted as an ASCII number, base 10.
2. If the number received in step 1 is non-zero, it is interpreted as the port number of a secondary stream to be used for the `stderr`. A second connection is then created to the specified port on the client's machine.
3. A null terminated user name of at most 16 characters is retrieved on the initial socket.
4. A null terminated password of at most 16 characters is retrieved on the initial socket.
5. A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list.
6. `rexecd` then validates the user as is done at login time and, if the authentication was successful, changes to the user's home directory, and establishes the user and group protections of the user. If any of these steps fail the connection is aborted and a diagnostic message is returned.
7. A null byte is returned on the connection associated with the `stderr` and the command line is passed to the normal login shell of the user. The shell inherits the network connections established by `rexecd`.

使用法

`in.rexecd` and `rexecd` are IPv6–enabled. See `ip6(7P)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

`svcs(1)`, `inetd(1M)`, `inetadm(1M)`, `svcadm(1M)`, `rexec(3SOCKET)`, `attributes(5)`, `smf(5)`, `ip6(7P)`

診断

All diagnostic messages are returned on the connection associated with the `stderr`, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 (0 is returned in step 7 above upon successful completion of all the steps prior to the command execution).

`username too long`	The name is longer than 16 characters.
`password too long`	The password is longer than 16 characters.

`command too long`	The command line passed exceeds the size of the argument list (as configured into the system).
`Login incorrect.`	No password file entry for the user name existed.
`Password incorrect.`	The wrong password was supplied.
`No remote directory.`	The `chdir` command to the home directory failed.
`Try again.`	A `fork` by the server failed.
`/usr/bin/sh: ...`	The user's login shell could not be started.

注意事項

The `in.rexecd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rexec:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

in.ripngd – network routing daemon for IPv6

形式

```
/usr/sbin/in.ripngd  [-s] [-q] [-t] [-p n] [-P] [-v ]
 [logfile]
```

機能説明

`in.ripngd` is the IPv6 equivalent of `in.routed(1M)`. It is invoked at boot time to manage the network routing tables. The routing daemon uses the Routing Information Protocol for IPv6.

`in.ripngd` is managed by the service management facility (SMF), by means of the service identifier:

```
svc:/network/routing/ripng:default
```

In normal operation, `in.ripngd` listens on the `udp(7P)` socket port 521 for routing information packets. If the host is an internetwork router, it periodically supplies copies of its routing tables to any directly connected hosts and networks.

When `in.ripngd` is started, it uses the `SIOCGLIFCONF` `ioctl(2)` to find those directly connected IPv6 interfaces configured into the system and marked "up"; the software loopback interface is ignored. If multiple interfaces are present, it is assumed the host will forward packets between networks. `in.ripngd` then multicasts a request packet on each IPv6 interface and enters a loop, listening for request and response packets from other hosts.

When a request packet is received, `in.ripngd` formulates a reply based on the information maintained in its internal tables. The response packet contains a list of known routes. With each route is a number specifying the number of bits in the prefix. The prefix is the number of bits in the high order part of an address that indicate the subnet or network that the route describes. Each route reported also has a "*hop count*" metric. A count of 16 or greater is considered "infinity." The metric associated with each route returned provides a metric relative to the sender.

The request packets received by `in.ripngd` are used to update the routing tables if one of the following conditions is satisfied:

- No routing table entry exists for the destination network or host, and the metric indicates the destination is "reachable", that is, the *hop count* is not infinite.
- The source host of the packet is the same as the router in the existing routing table entry. That is, updated information is being received from the very internetwork router through which packets for the destination are being routed.
- The existing entry in the routing table has not been updated for a period of time, defined to be 90 seconds, and the route is at least as cost-effective as the current route.
- The new route describes a shorter route to the destination than the one currently stored in the routing tables; this is determined by comparing the metric of the new route against the one stored in the table.

When an update is applied, `in.ripngd` records the change in its internal tables and generates a response packet to all directly connected hosts and networks. To allow possible unstable situations to settle, `in.ripngd` waits a short period of time (no more than 30 seconds) before modifying the kernel's routing tables.

In addition to processing incoming packets, `in.ripngd` also periodically checks the routing table entries. If an entry has not been updated for 3 minutes, the entry's metric is set to infinity and marked for deletion. Deletions are delayed an additional 60 seconds to insure the invalidation is propagated throughout the internet.

Hosts acting as internetwork routers gratuitously supply their routing tables every 30 seconds to all directly connected hosts and networks.

オプション

`in.ripngd` supports the options listed below. Listed with the options are the equivalent SMF property values. These are set for the `ripng:default` service with a command of the form:

```
# routeadm -m ripng:default key=value
```

`-p` *n* Send and receive the routing packets from other routers using the UDP port number *n*. Use of this option is equivalent to setting the `udp_port` property.

`-P` Do not use poison reverse. Use of this option is equivalent to setting the `poison_reverse` property to false.

`-q` Do not supply routing information. Use of this option is equivalent to setting the `quiet_mode` property to true.

`-s` Force `in.ripngd` to supply routing information whether it is acting as an internetwork router or not. Use of this option is equivalent to setting the `supply_routes` property to true.

`-t` Print all packets sent or received to standard output. `in.ripngd` will not divorce itself from the controlling terminal. Accordingly, interrupts from the keyboard will kill the process. Not supported by the `ripng` service.

`-v` Print all changes made to the routing tables to standard output with a timestamp. Use of this option is equivalent to setting the `verbose` property to true.

Any other argument supplied to this option is interpreted as the name of the file in which the actions of `in.ripngd`, as specified by this option or by `-t`, should be logged instead of being sent to standard output.

The logfile can be specified for the `ripng` service by means of the `log_file` property.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/routing

関連項目

in.routed(1M), routeadm(1M), svcadm(1M), ioctl(2), attributes(5), smf(5), udp(7P)

G. Malkin, R. Minnear, RFC 2080, RIPng for IPv6, January 1997.

注意事項

The kernel's routing tables may not correspond to those of in.ripngd for short periods of time while processes that utilize existing routes exit; the only remedy for this is to place the routing process in the kernel.

in.ripngd currently does not support all of the functionality of in.routed(1M). Future releases may support more if appropriate.

in.ripngd initially obtains a routing table by examining the interfaces configured on a machine. It then sends a request on all directly connected networks for more routing information. in.ripngd does not recognize or use any routing information already established on the machine prior to startup. With the exception of interface changes, in.ripngd does not see any routing table changes that have been done by other programs on the machine, for example, routes added, deleted or flushed by means of the route(1M) command. Therefore, these types of changes should not be done while in.ripngd is running. Rather, shut down in.ripngd, make the changes required, and then restart in.ripngd.

名前

in.rlogind, rlogind – remote login server

形式

```
/usr/sbin/in.rlogind [-k5eExXciPp] [-s tos] [-S keytab]
     [-M realm]
```

機能説明

`in.rlogind` is the server for the `rlogin(1)` program. The server provides a remote login facility with authentication based on Kerberos V5 or privileged port numbers.

`in.rlogind` is invoked by `inetd(1M)` when a remote login connection is established. When Kerberos V5 authentication is required (see option `-k` below), the authentication sequence is as follows:

- Check Kerberos V5 authentication.
- Check authorization according to the rules in `krb5_auth_rules(5)`.
- Prompt for a password if any checks fail and `/etc/pam.conf` is configured to do so.

In order for Kerberos authentication to work, a `host/<FQDN>` Kerberos principal must exist for each Fully Qualified Domain Name associated with the `in.rlogind` server. Each of these `host/<FQDN>` principals must have a `keytab` entry in the `/etc/krb5/krb5.keytab` file on the `in.rlogind` server. An example principal might be:

```
host/bigmachine.eng.example.com
```

See `kadmin(1M)` or `gkadmin(1M)` for instructions on adding a principal to a `krb5.keytab` file. See 『Oracle Solaris の管理: セキュリティーサービス』 for a discussion of Kerberos authentication.

If Kerberos V5 authentication is not enabled, then the authentication procedure follows the standard `rlogin` protocol:

- The server checks the client's source port. If the port is not in the range 512-1023, the server aborts the connection.
- The server checks the client's source address. If an entry for the client exists in both `/etc/hosts` and `/etc/hosts.equiv`, a user logging in from the client is not prompted for a password. If the address is associated with a host for which no corresponding entry exists in `/etc/hosts`, the user is prompted for a password, regardless of whether or not an entry for the client is present in `/etc/hosts.equiv`. See `hosts(4)` and `hosts.equiv(4)`.

Once the source port and address have been checked, `in.rlogind` allocates a pseudo-terminal and manipulates file descriptors so that the slave half of the pseudo-terminal becomes the `stdin`, `stdout`, and `stderr` for a login process. The login process is an instance of the `login(1)` program, invoked with the `-r`.

The login process then proceeds with the `pam(3PAM)` authentication process. See `SECURITY` below. If automatic authentication fails, it reprompts the user to login.

The parent of the login process manipulates the master side of the pseudo-terminal, operating as an intermediary between the login process and the client instance of the `rlogin` program. In normal operation, a packet protocol is invoked to provide Ctrl-S and Ctrl-Q type facilities and propagate interrupt signals to the remote programs. The login process propagates the client terminal's baud rate and terminal type, as found in the environment variable, `TERM`.

オプション

The following options are supported:

`-5`	Same as `-k`, for backwards compatibility.
`-c`	Requires Kerberos V5 clients to present a cryptographic checksum of initial connection information like the name of the user that the client is trying to access in the initial authenticator. This checksum provides additionl security by preventing an attacker from changing the initial connection information. This option is mutually exclusive with the `-i` option.
`-e`	Creates an encrypted session.
`-E`	Same as `-e`, for backwards compatibility.
`-i`	Ignores authenticator checksums if provided. This option ignores authenticator checksums presented by current Kerberos clients to protect initial connection information. Option `-i` is the opposite of option `-c`.
`-k`	Allows Kerberos V5 authentication with the `.k5login` access control file to be trusted. If this authentication system is used by the client and the authorization check is passed, then the user is allowed to log in.
`-M` *realm*	Uses the indicated Kerberos V5 realm. By default, the daemon will determine its realm from the settings in the `krb5.conf(4)` file.
`-p`	Prompts for authentication only if other authentication checks fail.
`-P`	Prompts for a password in addition to other authentication methods.
`-s` *tos*	Sets the `IP TOS` option.
`-S` *keytab*	Sets the `KRB5` keytab file to use. The`/etc/krb5/krb5.keytab` file is used by default.
`-x`	Same as `-e`, for backwards compatibility.
`-X`	Same as `-e`, for backwards compatibility.

使用法

`rlogind` and `in.rlogind` are IPv6–enabled. See `ip6(7P)`. `IPv6` is not currently supported with Kerberos V5 authentication.

Typically, Kerberized `rlogin` service runs on port 543 (klogin) and Kerberized, encrypted `rlogin` service runs on port 2105 (eklogin). The corresponding FMRI entries are:

```
svc:/network/login:klogin (rlogin with kerberos)
svc:/network/login:eklogin (rlogin with kerberos and encryption)
```

セキュリティー

`in.rlogind` uses pam(3PAM) for authentication, account management, and session management. The PAM configuration policy, listed through `/etc/pam.conf`, specifies the modules to be used for `in.rlogind`. Here is a partial `pam.conf` file with entries for the `rlogin` command using the "rhosts" and UNIX authentication modules, and the UNIX account, session management, and password management modules.

```
rlogin          auth sufficient     pam_rhosts_auth.so.1
rlogin          auth requisite      pam_authtok_get.so.1
rlogin          auth required       pam_dhkeys.so.1
rlogin          auth required       pam_unix_auth.so.1

rlogin          account required    pam_unix_roles.so.1
rlogin          account required    pam_unix_projects.so.1
rlogin          account required    pam_unix_account.so.1

rlogin          session required    pam_unix_session.so.1
```

With this configuration, the server checks the client's source address. If an entry for the client exists in both `/etc/hosts` and `/etc/hosts.equiv`, a user logging in from the client is not prompted for a password. If the address is associated with a host for which no corresponding entry exists in `/etc/hosts`, the user is prompted for a password, regardless of whether or not an entry for the client is present in `/etc/hosts.equiv`. See hosts(4) and hosts.equiv(4).

When running a Kerberized rlogin service (with or without the encryption option), the pam service name that should be used is "`krlogin`".

If there are no entries for the `rlogin` service, then the entries for the "other" service will be used. If multiple authentication modules are listed, then the user may be prompted for multiple passwords. Removing the `pam_rhosts_auth.so.1` entry will disable the `/etc/hosts.equiv` and `~/.rhosts` authentication protocol and the user would always be forced to type the password. The *sufficient* flag indicates that authentication through the `pam_rhosts_auth.so.1` module is sufficient to authenticate the user. Only if this authentication fails is the next authentication module used.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

login(1), svcs(1), rlogin(1), gkadmin(1M), in.rshd(1M), inetadm(1M), inetd(1M), kadmin(1M), svcadm(1M), pam(3PAM), hosts(4), hosts.equiv(4), krb5.conf(4), pam.conf(4), attributes(5), environ(5), krb5_auth_rules(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5)

『Oracle Solaris の管理: セキュリティーサービス』

診断

All diagnostic messages are returned on the connection associated with the stderr, after which any network connections are closed. An error is indicated by a leading byte with a value of 1.

`Hostname for your address unknown.`	No entry in the host name database existed for the client's machine.
`Try again.`	A *fork* by the server failed.
`/usr/bin/sh: ...`	The user's login shell could not be started.

注意事項

The authentication procedure used here assumes the integrity of each client machine and the connecting medium. This is insecure, but it is useful in an “open" environment.

A facility to allow all data exchanges to be encrypted should be present.

The in.rlogind service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/login:rlogin (rlogin)
svc:/network/login:klogin (rlogin with kerberos)
svc:/network/login:eklogin (rlogin with kerberos and encryption)
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

in.routed, routed – network routing daemon

形式

```
/usr/sbin/in.routed [-AdghmnqsStVz] [-T tracefile [-v]]
 [-F net[/mask ][,metric]] [-P params]
```

機能説明

The daemon `in.routed`, often referred to as `routed`, is invoked at boot time to manage the network routing tables. It uses Routing Information Protocol, RIPv1 (RFC 1058), RIPv2 (RFC 2453), and Internet Router Discovery Protocol (RFC 1256) to maintain the kernel routing table. The RIPv1 protocol is based on the reference 4.3BSD daemon.

`in.routed` is managed by means of the service management facility (SMF), using the fault management resource identifier (FMRI):

```
svc:/network/routing/route:default
```

The daemon listens on a `udp` socket for the `route` service (see `services(4)`) for Routing Information Protocol packets. It also sends and receives multicast Router Discovery ICMP messages. If the host is a router, `in.routed` periodically supplies copies of its routing tables to any directly connected hosts and networks. It also advertises or solicits default routes using Router Discovery ICMP messages.

When started (or when a network interface is later turned on), `in.routed` uses an `AF_ROUTE` address family facility to find those directly connected interfaces configured into the system and marked “up”. It adds necessary routes for the interfaces to the kernel routing table. Soon after being first started, and provided there is at least one interface on which RIP has not been disabled, `in.routed` deletes all pre-existing non-static routes in the kernel table. Static routes in the kernel table are preserved and included in RIP responses if they have a valid RIP metric (see `route(1M)`).

If more than one interface is present (not counting the loopback interface), it is assumed that the host should forward packets among the connected networks. After transmitting a RIP request and Router Discovery Advertisements or Solicitations on a new interface, the daemon enters a loop, listening for RIP request and response and Router Discovery packets from other hosts.

When a request packet is received, `in.routed` formulates a reply based on the information maintained in its internal tables. The response packet generated contains a list of known routes, each marked with a “hop count” metric (a count of 16 or greater is considered “infinite”). Advertised metrics reflect the metric associated with an interface (see `ifconfig(1M)`), so setting the metric on an interface is an effective way to steer traffic.

Responses do not include routes with a first hop on the requesting network, to implement in part split-horizon. Requests from query programs such as `rtquery(1M)` are answered with the complete table.

The routing table maintained by the daemon includes space for several gateways for each destination to speed recovery from a failing router. RIP response packets received are used to

update the routing tables, provided they are from one of the several currently recognized gateways or advertise a better metric than at least one of the existing gateways.

When an update is applied, `in.routed` records the change in its own tables and updates the kernel routing table if the best route to the destination changes. The change in the kernel routing table is reflected in the next batch of response packets sent. If the next response is not scheduled for a while, a flash update response containing only recently changed routes is sent.

In addition to processing incoming packets, in.routed also periodically checks the routing table entries. If an entry has not been updated for 3 minutes, the entry's metric is set to infinity and marked for deletion. Deletions are delayed until the route has been advertised with an infnite metric to insure the invalidation is propagated throughout the local internet. This is a form of poison reverse.

Routes in the kernel table that are added or changed as a result of ICMP Redirect messages are deleted after a while to minimize black-holes. When a TCP connection suffers a timeout, the kernel tells `in.routed`, which deletes all redirected routes through the gateway involved, advances the age of all RIP routes through the gateway to allow an alternate to be chosen, and advances of the age of any relevant Router Discovery Protocol default routes.

Hosts acting as internetwork routers gratuitously supply their routing tables every 30 seconds to all directly connected hosts and networks. These RIP responses are sent to the broadcast address on nets that support broadcasting, to the destination address on point-to-point links, and to the router's own address on other networks. If RIPv2 is enabled, multicast packets are sent on interfaces that support multicasting.

If no response is received on a remote interface, if there are errors while sending responses, or if there are more errors than input or output (see `netstat(1M)`), then the cable or some other part of the interface is assumed to be disconnected or broken, and routes are adjusted appropriately.

The Internet Router Discovery Protocol is handled similarly. When the daemon is supplying RIP routes, it also listens for Router Discovery Solicitations and sends Advertisements. When it is quiet and listening to other RIP routers, it sends Solicitations and listens for Advertisements. If it receives a good Advertisement and it is not multi-homed, it stops listening for broadcast or multicast RIP responses. It tracks several advertising routers to speed recovery when the currently chosen router dies. If all discovered routers disappear, the daemon resumes listening to RIP responses. It continues listening to RIP while using Router Discovery if multi-homed to ensure all interfaces are used.

The Router Discovery standard requires that advertisements have a default “lifetime” of 30 minutes. That means should something happen, a client can be without a good route for 30 minutes. It is a good idea to reduce the default to 45 seconds using `-P rdisc_interval=45` on the command line or `rdisc_interval=45` in the `/etc/gateways` file. See `gateways(4)`.

While using Router Discovery (which happens by default when the system has a single network interface and a Router Discover Advertisement is received), there is a single default route and a variable number of redirected host routes in the kernel table. On a host with more than one network interface, this default route will be via only one of the interfaces. Thus, multi-homed hosts running with `-q` might need the `no_rdisc` argument described below.

To support "legacy" systems that can handle neither RIPv2 nor Router Discovery, you can use the `pm_rdisc` parameter in the `/etc/gateways`. See `gateways(4)`.

By default, neither Router Discovery advertisements nor solicitations are sent over point-to-point links (for example, PPP). The Solaris OE uses a netmask of all ones (255.255.255.255) on point-to-point links.

`in.routed` supports the notion of "distant" passive or active gateways. When the daemon is started, it reads the file `/etc/gateways` to find such distant gateways that cannot be located using only information from a routing socket, to discover if some of the local gateways are passive, and to obtain other parameters. Gateways specified in this manner should be marked passive if they are not expected to exchange routing information, while gateways marked active should be willing to exchange RIP packets. Routes through passive gateways are installed in the kernel's routing tables once upon startup and are not included in transmitted RIP responses.

Distant active gateways are treated like network interfaces. RIP responses are sent to the distant active gateway. If no responses are received, the associated route is deleted from the kernel table and RIP responses are advertised via other interfaces. If the distant gateway resumes sending RIP responses, the associated route is restored.

Distant active gateways can be useful on media that do not support broadcasts or multicasts but otherwise act like classic shared media, such as some ATM networks. One can list all RIP routers reachable on the HIPPI or ATM network in `/etc/gateways` with a series of "host" lines. Note that it is usually desirable to use RIPv2 in such situations to avoid generating lists of inferred host routes.

Gateways marked external are also passive, but are not placed in the kernel routing table, nor are they included in routing updates. The function of external entries is to indicate that another routing process will install such a route if necessary, and that other routes to that destination should not be installed by `in.routed`. Such entries are required only when both routers might learn of routes to the same destination.

オプション

Listed below are available options. Any other argument supplied is interpreted as the name of a file in which the actions of `in.routed` should be logged. It is better to use `-T` (described below) instead of appending the name of the trace file to the command. Associated SMF properties for these options are described, and can be set by means of a command of the form:

```
# routeadm -m route:default name=value
```

`-A`

Do not ignore RIPv2 authentication if we do not care about RIPv2 authentication. This option is required for conformance with RFC 2453. However, it makes no sense and breaks using RIP as a discovery protocol to ignore all RIPv2 packets that carry authentication when this machine does not care about authentication. This option is equivalent to setting the `ignore_auth` property value to false.

`-d`

Do not run in the background. This option is meant for interactive use and is not usable under the SMF.

`-F net[/`*mask*`][,`*metric*`]`

Minimize routes in transmissions via interfaces with addresses that match *net* (network number)/*mask* (netmask), and synthesizes a default route to this machine with the *metric*. The intent is to reduce RIP traffic on slow, point-to-point links, such as PPP links, by replacing many large UDP packets of RIP information with a single, small packet containing a "fake" default route. If *metric* is absent, a value of 14 is assumed to limit the spread of the "fake" default route. This is a dangerous feature that, when used carelessly, can cause routing loops. Notice also that more than one interface can match the specified network number and mask. See also `-g`. Use of this option is equivalent to setting the `minimize_routes` property.

`-g`

Used on internetwork routers to offer a route to the "default" destination. It is equivalent to `-F 0/0,1` and is present mostly for historical reasons. A better choice is `-P pm_rdisc` on the command line or `pm_rdisc` in the `/etc/gateways` file. A larger metric will be used with the latter alternatives, reducing the spread of the potentially dangerous default route. The `-g` (or `-P`) option is typically used on a gateway to the Internet, or on a gateway that uses another routing protocol whose routes are not reported to other local routers. Note that because a metric of 1 is used, this feature is dangerous. Its use more often creates chaos with a routing loop than solves problems. Use of this option is equivalent to setting the `offer_default_route` property to true.

`-h`

Causes host or point-to-point routes not to be advertised, provided there is a network route going the same direction. That is a limited kind of aggregation. This option is useful on gateways to LANs that have other gateway machines connected with point-to-point links such as SLIP. Use of this option is equivalent to setting the `advertise_host_routes` property to false.

`-m`

Cause the machine to advertise a host or point-to-point route to its primary interface. It is useful on multi-homed machines such as NFS servers. This option should not be used except when the cost of the host routes it generates is justified by the popularity of the server. It is effective only when the machine is supplying routing information, because there is more than one interface. The `-m` option overrides the `-q` option to the limited extent

of advertising the host route. Use of this option is equivalent to setting the `advertise_host_routes_primary` property to true.

`-n`
Do not install routes in kernel. By default, routes are installed in the kernel. Use of this option is equivalent to setting the `install_routes` property to false.

`-P` *params*
Equivalent to adding the parameter line *params* to the `/etc/gateways` file. Can also be set by means of the `parameters` property.

`-q`
Opposite of the `-s` option. This is the default when only one interface is present. With this explicit option, the daemon is always in "quiet mode" for RIP and does not supply routing information to other computers. Use of this option is equivalent to setting the `quiet_mode` property to true.

`-s`
Force `in.routed` to supply routing information. This is the default if multiple network interfaces are present on which RIP or Router Discovery have not been disabled, and if global IPv4 forwarding is turned on (by means of `ipadm(1M)`). Use of this option is equivalent to setting the `supply_routes` property to true.

`-S`
If `in.routed` is not acting as an internetwork router, instead of entering the whole routing table in the kernel, it enters only a default route for each internetwork router. This reduces the memory requirements without losing any routing reliability. This option is provided for compatibility with the previous, RIPv1–only `in.routed`. Use of this option is generally discouraged. Use of this option is equivalent to setting the `default_routes_only` property to true.

`-t`
Runs in the foreground (as with `-d`) and logs the contents of the packets received (as with `-zz`). This is for compatibility with prior versions of Solaris and has no SMF equivalent.

`-T` *tracefile*
Increases the debugging level to at least 1 and causes debugging information to be appended to the trace file. Because of security concerns, do not to run `in.routed` routinely with tracing directed to a file. Use of this option is equivalent to setting the `log_file` property to `trace file path`.

`-v`
Enables debug. Similar to `-z`, except, where `-z` increments `trace_level`, `-v` sets `trace_level` to 1. Also, `-v` requires the `-T` option. Use of this option is equivalent to setting the `debug` property to true.

`-V`
Displays the version of the daemon.

`-z`
Increase the debugging level, which causes more information to be logged on the tracefile specified with `-T` or stdout. The debugging level can be increased or decreased with the `SIGUSR1` or `SIGUSR2` signals or with the `rtquery(1M)` command.

ファイル

`/etc/defaultrouter`	If this file is present and contains the address of a default router, the system startup script does not run `in.routed`. See `defaultrouter(4)`.
`/etc/gateways`	List of distant gateways and general configuration options for `in.routed`. See `gateways(4)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/routing

関連項目

`ipadm(1M)`, `route(1M)`, `routeadm(1M)`, `rtquery(1M)`, `svcadm(1M)`, `ioctl(2)`, `inet(3SOCKET)`, `defaultrouter(4)`, `gateways(4)`, `attributes(5)`, `icmp(7P)`, `inet(7P)`, `udp(7P)`

Internet Transport Protocols, XSIS 028112, Xerox System Integration Standard

Routing Information Protocol, v2 (RFC 2453, STD 0056, November 1998)

RIP-v2 MD5 Authentication (RFC 2082, January 1997)

Routing Information Protocol, v1 (RFC 1058, June 1988)

ICMP Router Discovery Messages (RFC 1256, September 1991)

注意事項

In keeping with its intended design, this daemon deviates from RFC 2453 in two notable ways:

- By default, `in.routed` does not discard authenticated RIPv2 messages when RIP authentication is not configured. There is little to gain from dropping authenticated packets when RIPv1 listeners will gladly process them. Using the `-A` option causes `in.routed` to conform to the RFC in this case.
- Unauthenticated RIP requests are never discarded, even when RIP authentication is configured. Forwarding tables are not secret and can be inferred through other means such as test traffic. RIP is also the most common router-discovery protocol, and hosts need to send queries that will be answered.

`in.routed` does not always detect unidirectional failures in network interfaces, for example, when the output side fails.

名前

in.rshd, rshd – remote shell server

形式

```
in.rshd [-k5eciU] [-s tos] [-S keytab] [-M realm]
     [-L env_var] host.port
```

機能説明

`in.rshd` is the server for the `rsh(1)` program. The server provides remote execution facilities with authentication based on Kerberos V5 or privileged port numbers.

`in.rshd` is invoked by `inetd(1M)` each time a shell service is requested.

When Kerberos V5 authentication is required (this can be set with Kerberos-specific options listed below), the following protocol is initiated:

1. Check Kerberos V5 authentication.
2. Check authorization according to rules in `krb5_auth_rules(5)`.
3. A null byte is returned on the initial socket and the command line is passed to the normal login shell of the user. (The `PATH` variable is set to `/usr/bin`.) The shell inherits the network connections established by `in.rshd`.

In order for Kerberos authentication to work, a `host/<`*FQDN*`>` Kerberos principal must exist for each Fully Qualified Domain Name associated with the `in.rshd` server. Each of these `host/<`*FQDN*`>` principals must have a `keytab` entry in the `/etc/krb5/krb5.keytab` file on the `in.rshd` server. An example principal might be:

```
host/bigmachine.eng.example.com
```

See `kadmin(1M)` or `gkadmin(1M)` for instructions on adding a principal to a `krb5.keytab` file. See 『Oracle Solaris の管理: セキュリティーサービス』 for a discussion of Kerberos authentication.

If Kerberos V5 authentication is not enabled, then `in.rshd` executes the following protocol:

1. The server checks the client's source port. If the port is not in the range 512-1023, the server aborts the connection. The client's host address (in hex) and port number (in decimal) are the arguments passed to `in.rshd`.
2. The server reads characters from the socket up to a null (`\\0`) byte. The resultant string is interpreted as an ASCII number, base 10.
3. If the number received in step 2 is non-zero, it is interpreted as the port number of a secondary stream to be used for the `stderr`. A second connection is then created to the specified port on the client's machine. The source port of this second connection is also in the range 512-1023.
4. A null-terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as the user identity on the *client*'s machine.
5. A null terminated user name of at most 16 characters is retrieved on the initial socket. This user name is interpreted as a user identity to use on the *server*'s machine.

6. A null terminated command to be passed to a shell is retrieved on the initial socket. The length of the command is limited by the upper bound on the size of the system's argument list.
7. `in.rshd` then validates the user according to the following steps. The remote user name is looked up in the password file and a `chdir` is performed to the user's home directory. If the lookup fails, the connection is terminated. If the `chdir` fails, it does a `chdir` to `/` (root). If the user is not the superuser, (user ID 0), and if the pam_rhosts_auth PAM module is configured for authentication, the file `/etc/hosts.equiv` is consulted for a list of hosts considered "equivalent". If the client's host name is present in this file, the authentication is considered successful. See the SECURITY section below for a discussion of PAM authentication.

   If the lookup fails, or the user is the superuser, then the file `.rhosts` in the home directory of the remote user is checked for the machine name and identity of the user on the client's machine. If this lookup fails, the connection is terminated
8. A null byte is returned on the initial connection and the command line is passed to the normal login shell of the user. The `PATH` variable is set to `/usr/bin`. The shell inherits the network connections established by `in.rshd`.

オプション

The following options are supported:

`-5`	Same as `-k`, for backwards compatibility
`-c`	Requires Kerberos V5 clients to present a cryptographic checksum of initial connection information like the name of the user that the client is trying to access in the initial authenticator. This checksum provides additionl security by preventing an attacker from changing the initial connection information. This option is mutually exclusive with the `-i` option.
`-e`	Requires the client to encrypt the connection.
`-i`	Ignores authenticator checksums if provided. This option ignores authenticator checksums presented by current Kerberos clients to protect initial connection information. Option `-i` is the opposite of option `-c`.
`-k`	Allows Kerberos V5 authentication with the `.k5login` access control file to be trusted. If this authentication system is used by the client and the authorization check is passed, then the user is allowed to log in.
`-L` *env_var*	List of environment variables that need to be saved and passed along.
`-M` *realm*	Uses the indicated Kerberos V5 realm. By default, the daemon will determine its realm from the settings in the `krb5.conf(4)` file.
`-s` *tos*	Sets the `IP TOS` option.
`-S` *keytab*	Sets the `KRB5` keytab file to use. The`/etc/krb5/krb5.keytab` file is used by default.

-U　Refuses connections that cannot be mapped to a name through the getnameinfo(3SOCKET) function.

使用法

rshd and in.rshd are IPv6–enabled. See ip6(7P). IPv6 is not currently supported with Kerberos V5 authentication.

The Kerberized rshd service runs on port 544 (kshell). The corresponding FMRI entry is: :

```
svc:/network/shell:kshell (rshd with kerberos (ipv4 only))
```

セキュリティー

in.rshd uses pam(3PAM) for authentication, account management, and session management. The PAM configuration policy, listed through /etc/pam.conf, specifies the modules to be used for in.rshd. Here is a partial pam.conf file with entries for the rsh command using rhosts authentication, UNIX account management, and session management module.

rsh	auth	required	pam_rhosts_auth.so.1
rsh	account	required	pam_unix_roles.so.1
rsh	session	required	pam_unix_projects.so.1
rsh	session	required	pam_unix_account.so.1
rsh	session	required	pam_unix_session.so.1

If there are no entries for the rsh service, then the entries for the “other” service are used. To maintain the authentication requirement for in.rshd, the rsh entry must always be configured with the pam_rhosts_auth.so.1 module.

in.rshd can authenticate using Kerberos V5 authentication or pam(3PAM). For Kerberized rsh service, the appropriate PAM service name is krsh.

ファイル

/etc/hosts.equiv

$HOME/.k5login　File containing Kerberos principals that are allowed access.

/etc/krb5/krb5.conf　Kerberos configuration file.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

rsh(1), svcs(1), gkadmin(1M), inetadm(1M), inetd(1M), kadmin(1M), svcadm(1M), pam(3PAM), getnameinfo(3SOCKET), hosts(4), krb5.conf(4), pam.conf(4), attributes(5), environ(5), krb5_auth_rules(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_rhosts_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5), ip6(7P)

『Oracle Solaris の管理: セキュリティーサービス』

診断

The following diagnostic messages are returned on the connection associated with stderr, after which any network connections are closed. An error is indicated by a leading byte with a value of 1 in step 8 above (0 is returned above upon successful completion of all the steps prior to the command execution).

locuser too long	The name of the user on the client's machine is longer than 16 characters.
remuser too long	The name of the user on the remote machine is longer than 16 characters.
command too long	The command line passed exceeds the size of the argument list (as configured into the system).
Hostname for your address unknown.	No entry in the host name database existed for the client's machine.
Login incorrect.	No password file entry for the user name existed.
Permission denied.	The authentication procedure described above failed.
Can’t make pipe.	The pipe needed for the stderr was not created.
Try again.	A *fork* by the server failed.

注意事項

The authentication procedure used here assumes the integrity of each client machine and the connecting medium. This is insecure, but it is useful in an “open” environment.

A facility to allow all data exchanges to be encrypted should be present.

The in.rshd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/shell:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

in.rwhod, rwhod – system status server

形式

/usr/sbin/in.rwhod [-m [*ttl*]]

機能説明

in.rwhod is the server which maintains the database used by the rwho(1) and ruptime(1) programs. Its operation is predicated on the ability to broadcast or multicast messages on a network.

in.rwhod operates as both a producer and consumer of status information. As a producer of information it periodically queries the state of the system and constructs status messages which are broadcast or multicast on a network. As a consumer of information, it listens for other in.rwhod servers' status messages, validating them, then recording them in a collection of files located in the directory /var/spool/rwho.

The rwho server transmits and receives messages at the port indicated in the rwho service specification, see services(4). The messages sent and received are defined in /usr/include/protocols/rwhod.h and are of the form:

```
struct  outmp {
        char     out_line[8];   /* tty name */
        char     out_name[8];   /* user id */
        long     out_time;      /* time on */
};
struct  whod {
        char     wd_vers;
        char     wd_type;
        char     wd_fill[2];
        int      wd_sendtime;
        int      wd_recvtime;
        char     wd_hostname[32];
        int      wd_loadav[3];
        int      wd_boottime;
        struct   whoent {
                 struct   outmp we_utmp;
                 int      we_idle;
     } wd_we[1024 / sizeof (struct whoent)];
 };
```

All fields are converted to network byte order prior to transmission. The load averages are as calculated by the w(1) program, and represent load averages over the 1, 5, and 15 minute intervals prior to a server's transmission. The host name included is that returned by the uname(2) system call. The array at the end of the message contains information about the users who are logged in to the sending machine. This information includes the contents of the utmpx(4) entry for each non-idle terminal line and a value indicating the time since a character was last received on the terminal line.

Messages received by the rwho server are discarded unless they originated at a rwho server's port. In addition, if the host's name, as specified in the message, contains any unprintable ASCII characters, the message is discarded. Valid messages received by in.rwhod are placed in files named whod.hostname in the directory /var/spool/rwho. These files contain only the most recent message, in the format described above.

Status messages are generated approximately once every 3 minutes.

オプション

The following options are supported:

-m [*ttl*] Use the rwho IP multicast address (224.0.1.3) when transmitting. Receive announcements both on this multicast address and on the IP broadcast address. If *ttl* is not specified in.rwhod multicasts on all interfaces but with the IP TimeToLive set to 1 (that is, packets are not forwarded by multicast routers.) If *ttl* is specified in.rwhod only transmits packets on one interface and setting the IP TimeToLive to the specified *ttl*.

ファイル

/var/spool/rwho/whod.* information about other machines

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

ruptime(1), rwho(1), w(1), uname(2), services(4), utmpx(4), attributes(5)

警告

This service can cause network performance problems when used by several hosts on the network. It is not run at most sites by default. If used, include the -m multicast option.

注意事項

This service takes up progressively more network bandwidth as the number of hosts on the local net increases. For large networks, the cost becomes prohibitive.

in.rwhod should relay status information between networks. People often interpret the server dying as a machine going down.

名前

install – install commands

形式

```
/usr/sbin/install -c dira [-m mode] [-u user] [-g group]
     [-o] [-s] file

/usr/sbin/install -f dirb [-m mode] [-u user] [-g group]
     [-o] [-s] file

/usr/sbin/install -n dirc [-m mode] [-u user] [-g group]
     [-o] [-s] file

/usr/sbin/install -d | -i [-m mode] [-u user] [-g group]
     [-o] [-s] dirx...

/usr/sbin/install [-m mode] [-u user] [-g group] [-o] [-s] file
     [dirx]...
```

機能説明

`install` is most commonly used in "makefiles" (see `make(1S)`) to install a `file` in specific locations, or to create directories within a file system. Each `file` is installed by copying it into the appropriate directory.

`install` uses no special privileges to copy files from one place to another. The implications of this are:

- You must have permission to read the files to be installed.
- You must have permission to copy into the destination directory.
- You must have permission to change the modes on the final copy of the file if you want to use the `-m` option.
- You must be super-user if you want to specify the ownership of the installed file with the `-u` or `-g` options. If you are not the super-user, the installed file is owned by you, regardless of who owns the original.

Note that if the `ROOT` environment variable is set, each of the default directory paths are prefixed by its value (for example, `$ROOT/bin` and so on).

`install` prints messages telling the user exactly what files it is replacing or creating and where they are going.

If no options or directories (*dirx* . . .) are given, `install` searches a set of default directories (`/bin`, `/usr/bin`, `/etc`, `/lib`, and `/usr/lib`, in that order) for a file with the same name as `file`. When the first occurrence is found, `install` issues a message saying that it is overwriting that file with `file`, and proceeds to do so. If the file is not found, the program states this and exits.

If one or more directories (*dirx* . . .) are specified after `file`, those directories are searched before the default directories.

This version of `install` (`/usr/sbin/install`) is not compatible with the `install` binaries in many versions of Unix other than Solaris. For a higher degree of compatibility with other Unix versions, use `/usr/ucb/install`, which is described in the `install(1B)` man page.

オプション

The following options are supported:

- `-c` *dira* — Install `file` in the directory specified by *dira*, if `file` does not yet exist. If it is found, `install` issues a message saying that the file already exists, and exits without overwriting it.
- `-f` *dirb* — Force `file` to be installed in given directory, even if the file already exists. If the file being installed does not already exist, the mode and owner of the new file is set to `755` and `bin` , respectively. If the file already exists, the mode and owner is that of the already existing file.
- `-n` *dirc* — If `file` is not found in any of the searched directories, it is put in the directory specified in *dirc*. The mode and owner of the new file is set to `755` and `bin`, respectively.
- `-d` — Create a directory. Missing parent directories are created as required as in `mkdir -p`. If the directory already exists, the owner, group and mode is set to the values given on the command line.
- `-i` — Ignore default directory list, searching only through the given directories (*dirx* . . .).
- `-m` *mode* — The mode of the new file is set to *mode*. Set to `0755` by default.
- `-u` *user* — The owner of the new file is set to *user*. Only available to the super-user. Set to `bin` by default.
- `-g` *group* — The group id of the new file is set to *group*. Only available to the super-user. Set to `bin` by default.
- `-o` — If `file` is found, save the “found" file by copying it to `OLDfile` in the directory in which it was found. This option is useful when installing a frequently used file such as `/bin/sh` or `/lib/saf/ttymon`, where the existing file cannot be removed.
- `-s` — Suppress printing of messages other than error messages.

使用法

See `largefile(5)` for the description of the behavior of `install` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`chgrp(1)`, `chmod(1)`, `chown(1)`, `cp(1)`, `install(1B)`, `make(1S)`, `mkdir(1)`, `attributes(5)`, `largefile(5)`

名前

installboot – ディスクパーティションへのブートブロックのインストール

形式

`installboot [-F zfs|ufs|hsfs]` *bootblk raw-disk-device*

機能説明

boot(1M) プログラム `ufsboot` は、ディスクパーティションのブート領域に置かれているブートブロックプログラムによってディスクから読み込まれます。このプログラムはファイルシステムに固有で、ブートするディスク上のファイルシステムのタイプと一致している必要があります。

ブートオブジェクトはプラットフォームに依存し、`/usr/platform/`*platform-name*`/lib/fs/`*file-system* ディレクトリに置かれています。プラットフォーム名を調べるには、uname(1) の `-i` オプションを使用します。ファイルシステムタイプを調べるには、次を使用します。

```
% fstyp raw-disk-device
```

fstyp(1M) を参照してください。

`installboot` ユーティリティーは SPARC 専用のプログラムです。x86 アーキテクチャーではサポートされていません。x86 ユーザーは代わりに installgrub(1M) を使用してください。

オプション

次のオプションがサポートされています。

`-F zfs|ufs|hsfs`
: インストールするブートブロックのファイルシステムタイプを指定します。`zfs` または `hsfs` を指定する場合には必須です。デフォルトは `ufs` です。

オペランド

bootblk
: ブートブロックコードの名前。

raw-disk-device
: ブートブロックコードをインストールするディスクデバイスの名前。これは、読み書き可能な文字デバイスである必要があります。命名規約は、SCSI または IPI ドライブの場合は `c?t ?d?s?`、IDE ドライブの場合は `c?d?s?` です。

使用例

例 1　UFS ブートブロックのインストール

コマンドを実行するプラットフォームのコントローラ 1 のターゲット 0 のスライス 0 に `ufs` ブートブロックをインストールするには、次を使用します。

```
# installboot /usr/platform/‘uname -i‘/lib/fs/ufs/bootblk \
    /dev/rdsk/c1t0d0s0
```

例 2　ZFS ブートブロックのインストール

コマンドを実行するプラットフォームのコントローラ 1 のターゲット 0 のスライス 0 に ZFS ブートブロックをインストールするには、次のような構文を使用します。

```
# installboot -F zfs /usr/platform/‘uname -i‘/lib/fs/zfs/bootblk \
      /dev/rdsk/c1t1d0s0
```

ファイル

/usr/platform/*platform-name*/lib/fs/ ブートオブジェクトが置かれているディレクトリ。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

kmdb(1), od(1), uname(1), boot(1M), fstyp(1M), init(1M), kernel(1M), monitor(1M), reboot(1M), rpc.bootparamd(1M), init.d(4), attributes(5)

『Oracle Solaris 11 システムのインストール』

警告

bootblk ファイルまたは *openfirmware* ファイルが存在しない場合、あるいは raw ディスクデバイスが文字デバイスでない場合、installboot ユーティリティーは失敗します。

名前

installf – add a file to the software installation database

形式

```
installf [-c class] [ [-M] -R root_path] [-V fs_file] pkginst pathname
      [ftype [major minor] [mode owner group]]

installf [-c class] [ [-M] -R root_path] [-V fs_file] pkginst –

installf -f [-c class] [ [-M] -R root_path] [-V fs_file] pkginst
```

機能説明

`installf` informs the system that a pathname not listed in the `pkgmap(4)` file is being created or modified. It should be invoked before any file modifications have occurred.

When the second synopsis is used, the pathname descriptions will be read from standard input. These descriptions are the same as would be given in the first synopsis but the information is given in the form of a list. The descriptions should be in the form:

pathname [*ftype* [*major minor*] [*mode owner group*]]

After all files have been appropriately created and/or modified, `installf` should be invoked with the `-f` synopsis to indicate that installation is final. Links will be created at this time and, if attribute information for a pathname was not specified during the original invocation of `installf`, or was not already stored on the system, the current attribute values for the pathname will be stored. Otherwise, `installf` verifies that attribute values match those given on the command line, making corrections as necessary. In all cases, the current content information is calculated and stored appropriately.

Package commands are `largefile(5)`-aware. They handle files larger than 2 GB in the same way they handle smaller files. In their current implementations, `pkgadd(1M)`, `pkgtrans(1)` and other package commands can process a datastream of up to 4 GB.

オプション

`-c` *class*	Class to which installed objects should be associated. Default class is `none`.
`-f`	Indicates that installation is complete. This option is used with the final invocation of `installf` (for all files of a given class).
`-M`	Instruct `installf` not to use the $*root_path*`/etc/vfstab` file for determining the client's mount points. This option assumes the mount points are correct on the server and it behaves consistently with Solaris 2.5 and earlier releases.
`-R` *root_path*	Define the full path name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*. The *root_path* can be specified when installing to a client from a server (for example, `/export/root/client1`). `installf` inherits the value of the `PKG_INSTALL_ROOT` environment variable. (See ENVIRONMENT VARIABLES, below.) If `PKG_INSTALL_ROOT` is set, such as when the `-R` option is used with `pkgadd(1M)` or `pkgrm(1M)`

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-V` *fs_file* Specify an alternative *fs_file* to map the client's file systems. For example, used in situations where the `$`*root_path*`/etc/vfstab` file is non-existent or unreliable.

オペランド

pkginst Name of package instance with which the pathname should be associated.

pathname Pathname that is being created or modified.

ftype A one-character field that indicates the file type. Possible file types include:

- `b` block special device
- `c` character special device
- `d` directory
- `e` a file to be edited upon installation or removal
- `f` a standard executable or data file
- `l` linked file
- `p` named pipe
- `s` symbolic link
- `v` volatile file (one whose contents are expected to change)
- `x` an exclusive directory

major The major device number. The field is only specified for block or character special devices.

minor The minor device number. The field is only specified for block or character special devices.

mode The octal mode of the file (for example, 0664). A question mark (`?`) indicates that the mode will be left unchanged, implying that the file already exists on the target machine. This field is not used for linked or symbolically linked files.

owner The owner of the file (for example, `bin` or `root`). The field is limited to 14 characters in length. A question mark (`?`) indicates that the owner will be left unchanged, implying that the file already exists on the target machine. This field is not used for linked or symbolically linked files.

group The group to which the file belongs (for example, `bin` or `sys`). The field is limited to 14 characters in length. A question mark (`?`) indicates that the group will be left unchanged, implying that the file already exists on the target

machine. This field is not used for linked or symbolically linked files.

使用例

例 1 Basic Usage

The following example shows the use of `installf`, invoked from an optional pre-install or post-install script:

```
# create /dev/xt directory
# (needs to be done before drvinstall)
installf $PKGINST /dev/xt d 755 root sys ||
    exit 2
majno=‘/usr/sbin/drvinstall -m /etc/master.d/xt
     -d $BASEDIR/data/xt.o -v1.0‘ ||
    exit 2
i=00
while [ $i –lt $limit ]
do
    for j in 0 1 2 3 4 5 6 7
    do
        echo /dev/xt$i$j c $majno ‘expr $i ? 8 + $j‘
             644 root sys |
        echo /dev/xt$i$j=/dev/xt/$i$j
    done
    i=‘expr $i + 1‘
    [ $i –le 9 ] && i="0$i" #add leading zero
done | installf $PKGINST – || exit 2
# finalized installation, create links
installf -f $PKGINST || exit 2
```

環境変数

`installf` inherits the value of the following environment variable. This variable is set when `pkgadd(1M)` or `pkgrm(1M)` is invoked with the `-R` option.

`PKG_INSTALL_ROOT` If present, defines the full path name of a directory to use as the system's `PKG_INSTALL_ROOT` path. All product and package information files are then looked for in the directory tree, starting with the specified `PKG_INSTALL_ROOT` path. If not present, the default system path of `/` is used.

終了ステータス

`0` Successful operation.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

pkginfo(1), pkgmk(1), pkgparam(1), pkgproto(1), pkgtrans(1), pkgadd(1M), pkgask(1M), pkgchk(1M), pkgrm(1M), removef(1M), pkgmap(4), space(4), attributes(5), largefile(5)

『Application Packaging Developer's Guide』

注意事項

When *ftype* is specified, all applicable fields, as shown below, must be defined:

ftype	Required Fields
p, x, d, f, v, or e	mode owner group
c or b	major minor mode owner group

The installf command will create directories, named pipes and special devices on the original invocation. Links are created when installf is invoked with the -f option to indicate installation is complete.

Links should be specified as *path1=path2*. *path1* indicates the destination and *path2* indicates the source file.

Files installed with installf will be placed in the class none, unless a class is defined with the command. Subsequently, they will be removed when the associated package is deleted. If this file should not be deleted at the same time as the package, be certain to assign it to a class which is ignored at removal time. If special action is required for the file before removal, a class must be defined with the command and an appropriate class action script delivered with the package.

When classes are used, installf must be used in one of the following forms:

```
installf -c class1 . . .
installf -f -c class1 . . .
installf -c class2 . . .
installf -f -c class2 . . .
```

名前

installgrub – ディスクパーティションへの GRUB のインストール

形式

`/usr/sbin/installgrub [-fm]` *stage1 stage2 raw-device*

機能説明

`installgrub` コマンドは x86 専用のプログラムです。GRUB は GRand Unified Bootloader を表します。

`installgrub` は、GRUB のステージ 1 ファイルとステージ 2 ファイルを、特定のディスクパーティションのブート領域にインストールします。`-m` オプションが指定された場合、`installgrub` は、ステージ 1 ファイルをディスクのマスターブートセクターにインストールします。

オプション

`installgrub` コマンドで使用できるオプションは、次のとおりです。

`-f` マスターブートセクターの上書き時に対話を抑制します。

`-m` GRUB の *stage1* を、マスターブートセクター上に対話的にインストールします。Solaris が拡張パーティションにインストールされている場合は、このオプションを使用する必要があります。

オペランド

`installgrub` コマンドで使用できるオペランドは、次のとおりです。

stage1 GRUB のステージ 1 ファイルの名前。

stage2 GRUB のステージ 2 ファイルの名前。

raw-device GRUB コードのインストール先となるデバイスの名前。これは、読み書き可能な文字デバイスである必要があります。ディスクデバイスの場合、GRUB メニューファイルの格納先となるスライスを指定します。(Solaris の場合、これはルートスライスになる)。

使用例

例 1 ハードディスクスライスへの GRUB のインストール

次のコマンドは、ルートスライスが `c0d0s0` であるようなシステム上に、GRUB をインストールします。

```
example# /usr/sbin/installgrub /boot/grub/stage1 \
         /boot/grub/stage2 /dev/rdsk/c0d0s0
```

ファイル

`/boot/grub` GRUB ファイルが存在しているディレクトリ。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	不確実

関連項目 boot(1M), fdisk(1M), fmthard(1M), kernel(1M), attributes(5)

警告 マスターブートセクター上に GRUB をインストールする (-m オプション) と、マシン上に現在インストールされているブートマネージャーが上書きされます。どの fdisk パーティションがアクティブになっているかにかかわらず、システムは常に Solaris パーティション内の GRUB をブートするようになります。

名前

in.stdiscover – Service Tag Discovery Daemon

形式

`/usr/lib/inet/in.stdiscover`

機能説明

The `in.stdiscover` daemon allows a mechanism for discovering the location of the Service Tag Listener. By default, the `in.stdiscover` daemon listens for discovery probes (using a minimal built-in protocol) on UDP port 6481.

The daemon is under control of the service management facility, `smf(5)`, under its `inetd` framework. It only runs upon demand and exits when no longer in use.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/service-tag
Interface Stability	Private

関連項目

`in.stlisten(1M)`, `stclient(1M)`

名前

in.stlisten – Service Tag Listener

形式

`/usr/lib/inet/in.stlisten`

機能説明

The `in.stlisten` daemon allows a mechanism for discovering the location of the Service Tag. By default, the `in.stlisten` daemon listens for discovery probes (using a minimal built-in protocol) on TCP port 6481.

The daemon is under control of the service management facility, `smf(5)`, under its `inetd` framework. It only runs upon demand and exits when no longer in use.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/service-tag
Interface Stability	Private

関連項目

`in.stdiscover(1M)`, `stclient(1M)`, `svccfg(1M)`, `attributes(5)`, `environ(5)`, `smf(5)`

注意事項

In open networks where the participants may not always be trusted, it is recommended that you deploy this daemon with the `passphrase-encryption` option. In `smf(5)` environments, the following command can be used to set the *passphrase*:

```
< prepare a text file "passfile" containing the passphrase >
# chown svctag:daemon passfile
# chmod 600 passfile
svccfg -s svc:/network/stlisten \
        setprop servicetag/passphrase=passfile
```

where *passfile* is the path of a file containing the intended *passphrase*.

This *passphrase* can be subsequently cleared as follows:

```
svccfg -s svc:/network/stlisten \
        setprop servicetag/passphrase=""
```

名前

in.talkd, talkd – server for talk program

形式

```
in.talkd
```

機能説明

talkd is a server used by the talk(1) program. It listens at the UDP port indicated in the "talk" service description; see services(4). The actual conversation takes place on a TCP connection that is established by negotiation between the two machines involved.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

svcs(1), talk(1), inetadm(1M), inetd(1M), svcadm(1M), services(4), attributes(5), smf(5)

注意事項

The protocol is architecture dependent.

The in.talkd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/talk
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

in.telnetd, telnetd – DARPA TELNET protocol server

形式

```
/usr/sbin/in.telnetd [-a authmode] [-EXUh] [-s tos]
     [-S keytab] [-M realm]
```

機能説明

`in.telnetd` is a server that supports the DARPA standard TELNET virtual terminal protocol. `in.telnetd` is normally invoked in the internet server (see `inetd(1M)`), for requests to connect to the TELNET port as indicated by the `/etc/services` file (see `services(4)`).

`in.telnetd` operates by allocating a pseudo-terminal device for a client, then creating a login process which has the slave side of the pseudo-terminal as its standard input, output, and error. `in.telnetd` manipulates the master side of the pseudo-terminal, implementing the TELNET protocol and passing characters between the remote client and the login process.

When a TELNET session starts up, `in.telnetd` sends TELNET options to the client side indicating a willingness to do *remote* `echo` of characters, and to *suppress go ahead*. The pseudo-terminal allocated to the client is configured to operate in "cooked" mode, and with XTABS, ICRNL and ONLCR enabled. See `termio(7I)`.

`in.telnetd` is willing to do: *echo*, *binary*, *suppress go ahead*, and *timing mark*. `in.telnetd` is willing to have the remote client do: *binary*, *terminal type*, *terminal size*, *logout option*, and *suppress go ahead*.

`in.telnetd` also allows environment variables to be passed, provided that the client negotiates this during the initial option negotiation. The DISPLAY environment variable may be sent this way, either by the TELNET general environment passing methods, or by means of the XDISPLOC TELNET option. DISPLAY can be passed in the environment option during the same negotiation where XDISPLOC is used. Note that if you use both methods, use the same value for both. Otherwise, the results may be unpredictable.

These options are specified in Internet standards RFC 1096, RFC 1408, RFC 1510, RFC 1571, RFC 2941, RFC 2942, RFC 2946, and RFC 1572. The following Informational draft is also supported: RFC 2952.

The banner printed by `in.telnetd` is configurable. The default is (more or less) equivalent to ‘`uname -sr`‘ and will be used if no banner is set in `/etc/default/telnetd`. To set the banner, add a line of the form

```
BANNER="..."
```

to `/etc/default/telnetd`. Nonempty banner strings are fed to shells for evaluation. The default banner may be obtained by

```
BANNER="\\r\\n\\r\\n‘uname -s‘ ‘uname -r‘\\r\\n\\r\\n"
```

and no banner will be printed if `/etc/default/telnetd` contains

```
BANNER=""
```

オプション

The following options are supported:

`-a` *authmode*
: This option may be used for specifying what mode should be used for authentication. There are several valid values for *authmode*:

  `valid`
  : Only allows connections when the remote user can provide valid authentication information to identify the remote user, and is allowed access to the specified account without providing a password.

  `user`
  : Only allows connections when the remote user can provide valid authentication information to identify the remote user. The `login(1)` command will provide any additional user verification needed if the remote user is not allowed automatic access to the specified account.

  `none`
  : This is the default state. Authentication information is not required. If no or insufficient authentication information is provided, then the `login`(1) program provides the necessary user verification.

  `off`
  : This disables the authentication code. All user verification happens through the `login`(1) program.

`-E`
: Disables encryption support negotiation.

`-h`
: Disables displaying host specific information before login has been completed.

`-M` *realm*
: Uses the indicated Kerberos V5 realm. By default, the daemon will determine its realm from the settings in the `krb5.conf(4)` file.

`-s` *tos*
: Sets the `IP TOS` option.

`-S` *keytab*
: Sets the `KRB5` keytab file to use. The`/etc/krb5/krb5.keytab` file is used by default.

`-U`
: Refuses connections that cannot be mapped to a name through the `getnameinfo(3SOCKET)` function.

`-X`
: Disables Kerberos V5 authentication support negotiation.

使用法

`telnetd` and `in.telnetd` are IPv6–enabled. See `ip6(7P)`.

セキュリティー

`in.telnetd` can authenticate using Kerberos V5 authentication, `pam(3PAM)`, or both. By default, the telnet server will accept valid Kerberos V5 authentication credentials from a `telnet` client that supports Kerberos. `in.telnetd` can also support an encrypted session from such a client if the client requests it.

The telnet protocol only uses single DES for session protection—clients request service tickets with single DES session keys. The KDC must know that host service principals that offer the telnet service support single DES, which, in practice, means that such principals must have single DES keys in the KDC database.

In order for Kerberos authentication to work, a host/<*FQDN*> Kerberos principal must exist for each Fully Qualified Domain Name associated with the telnetd server. Each of these host/<*FQDN*> principals must have a keytab entry in the /etc/krb5/krb5.keytab file on the telnetd server. An example principal might be:

```
host/bigmachine.eng.example.com
```

See kadmin(1M) or gkadmin(1M) for instructions on adding a principal to a krb5.keytab file. See 『Oracle Solaris の管理: セキュリティーサービス』 for a discussion of Kerberos authentication.

in.telnetd uses pam(3PAM) for authentication, account management, session management, and password management. The PAM configuration policy, listed through /etc/pam.conf, specifies the modules to be used for in.telnetd. Here is a partial pam.conf file with entries for the telnet command using the UNIX authentication, account management, session management, and password management modules.

```
telnet  auth requisite          pam_authtok_get.so.1
telent  auth required           pam_dhkeys.so.1
telent  auth required           pam_unix_auth.so.1

telnet  account requisite       pam_roles.so.1
telnet  account required        pam_projects.so.1
telnet  account required        pam_unix_account.so.1

telnet  session required        pam_unix_session.so.1

telnet  password required       pam_dhkeys.so.1
telent  password requisite      pam_authtok_get.so.1
telnet  password requisite      pam_authtok_check.so.1
telnet  password required       pam_authtok_store.so.1
```

If there are no entries for the telnet service, then the entries for the “other” service will be used. If multiple authentication modules are listed, then the user may be prompted for multiple passwords.

For a Kerberized telnet service, the correct PAM service name is ktelnet.

ファイル

- /etc/default/telnetd

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/telnet

関連項目

login(1), svcs(1), telnet(1), gkadmin(1M), inetadm(1M), inetd(1M), kadmin(1M), svcadm(1M), pam(3PAM), getnameinfo(3SOCKET), issue(4), krb5.conf(4), pam.conf(4), services(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5), ip6(7P), termio(7I)

『Oracle Solaris の管理: セキュリティーサービス』

Alexander, S. RFC 1572, TELNET Environment Option. Network Information Center, SRI International, Menlo Park, Calif., January 1994.

Borman, Dave. RFC 1408, TELNET Environment Option. Network Information Center, SRI International, Menlo Park, Calif., January 1993.

Borman, Dave. RFC 1571, TELNET Environment Option Interoperability Issues. Network Information Center, SRI International, Menlo Park, Calif., January 1994.

Crispin, Mark. RFC 727, TELNET Logout Option. Network Information Center, SRI International, Menlo Park, Calif., April 1977.

Marcy, G. RFC 1096, TELNET X Display Location Option. Network Information Center, SRI International, Menlo Park, Calif., March 1989.

Postel, Jon, and Joyce Reynolds. RFC 854, TELNET Protocol Specification. Network Information Center, SRI International, Menlo Park, Calif., May 1983.

Waitzman, D. RFC 1073, TELNET Window Size Option. Network Information Center, SRI International, Menlo Park, Calif., October 1988.

Kohl, J., Neuman, C., The Kerberos Network Authentication Service (V5), RFC 1510. September 1993.

Ts'o, T. and J. Altman, Telnet Authentication Option, RFC 2941. September 2000.

Ts'o, T., Telnet Authentication: Kerberos Version 5, RFC 2942. September 2000.

Ts'o, T., Telnet Data Encryption Option, RFC 2946. September 2000.

Ts'o, T., Telnet Encryption: DES 64 bit Cipher Feedback, RFC 2952. September 2000.

注意事項

Some TELNET commands are only partially implemented.

Binary mode has no common interpretation except between similar operating systems.

The terminal type name received from the remote client is converted to lower case.

The *packet* interface to the pseudo-terminal should be used for more intelligent flushing of input and output queues.

`in.telnetd` never sends TELNET *go ahead* commands.

The `in.telnetd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/telnet
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

in.tftpd, tftpd – Internet Trivial File Transfer Protocol server

形式

`in.tftpd [-d] [-T` *rexmtval*`] [-s] [`*homedir*`]`

機能説明

`tftpd` is a server that supports the Internet Trivial File Transfer Protocol (TFTP).

Before responding to a request, the server attempts to change its current directory to *homedir*; the default directory is `/tftpboot`.

The use of `tftp` does not require an account or password on the remote system. Due to the lack of authentication information, `in.tftpd` will allow only publicly readable files to be accessed. Files may be written only if they already exist and are publicly writable. Note that this extends the concept of "public" to include all users on all hosts that can be reached through the network. This may not be appropriate on all systems, and its implications should be considered before enabling this service.

`in.tftpd` runs with the user ID and group ID set to `[GU]ID_NOBODY` under the assumption that no files exist with that owner or group. However, nothing checks this assumption or enforces this restriction.

オプション

`-d`	Debug. When specified it sets the `SO_DEBUG` socket option.
`-s`	Secure. When specified, the directory change to *homedir* must succeed. The daemon also changes its root directory to *homedir*.
`-T` *rexmtval*	Specifies the value of the retransmission timeout in seconds. This also affects the maximum session timeout in that the latter is set to five times the retransmission timeout value.

使用法

The `in.tftpd` server is IPv6–enabled. See `ip6(7P)`.

`in.tftpd` supports transfers of greater than 32 MB, per RFC 2348.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/tftp

関連項目

`svcs(1)`, `tftp(1)`, `inetadm(1M)`, `inetd(1M)`, `svcadm(1M)`, `netconfig(4)`, `attributes(5)`, `smf(5)`, `ip6(7P)`

Malkin, G. and Harkin, A. RFC 2347, TFTP Option Extension. The Internet Society. May 1998

Malkin, G. and Harkin, A. RFC 2348, TFTP Blocksize Option. The Internet Society. May 1998

Malkin, G. and Harkin, A. RFC 2349, TFTP Timeout Interval and Transfer Size Options. The Internet Society. May 1998

Sollins, K.R. RFC 1350, The TFTP Protocol (Revision 2). Network Working Group. July 1992.

注意事項

The tftpd server only acknowledges the transfer size option that is sent with a read request when the octet transfer mode is specified.

The in.tftpd.1m service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/tftp/udp6:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

Unlike most smf(5) services, a manifest for the tftp service is not included in the system. To create one and enable this service, the administrator should:

1. Edit /etc/inet/inetd.conf and uncomment the tftp entry.
2. Run /usr/sbin/inetconv.

After you run inetconv, the svc:/network/tftp/udp6:default service is created and enabled.

名前

in.timed – UDP or TCP time protocol service daemon

形式

```
in.timed
```

```
FMRI svc:/internet/time:default
```

機能説明

FMRI stands for Fault Management Resource Identifier. It is used to identify resources managed by the Fault Manager. See fmd(1M) and smf(5).

The in.timed service provides the server-side of the time protocol. The time server sends to requestors the time in seconds since midnight, January 1, 1900. The time protocol is available on both TCP and UDP transports through port 37.

The in.timed service is an inetd(1M) smf(5) delegated service. The in.timed detects which transport is requested by examining the socket it is passed by the inetd daemon.

TCP-based service: Once a connection is established, the in.timed sends the time as a 32-bit binary number and closes the connection. Any received data is ignored.

UDP-based service: The in.timed listens for UDP datagrams. When a datagram is received, the server generates a UDP datagram containing the time as a 32–bit binary number and sends it to the client. Any received data is ignored.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/legacy-network-services
Interface Stability	Committed

関連項目

inetd(1M), attributes(5), smf(5)

RFC 868

名前

intrd – 割り込み分散デーモン

形式

```
intrd
```

機能説明

intrd デーモンはブート時に起動され、割り込みと CPU の間の割り当てを監視します。現在の割り当てが不均衡で、システムパフォーマンスに悪影響を与えると判断した場合、intrd は新しい割り当てを生成して実装します。

通知はすべて syslogd(1M) を介して配信されます。

intrd は、最適なパフォーマンスを得るためにシステムを動的に監視するため、アイドル状態のシステム上でも少量の CPU 時間を消費します。これは正常な動作です。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/kernel/power
インタフェースの安定性	不確実

関連項目

svcs(1), svcadm(1M), syslogd(1M), attributes(5), smf(5)

注意事項

割り込み分散デーモンは、サービス管理機能 smf(5) により次のサービス識別子の下で管理されます。

```
svc:/system/intrd:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

intrstat – 割り込み統計の報告

形式

```
/usr/sbin/intrstat [-c cpulist | -C processor_set_id] [-T u | d ]
     [-x opt[=val]] [interval [count]]
```

機能説明

intrstat ユーティリティーは、実行時の割り込み統計を収集して表示します。出力はデバイス名と CPU ID から成るテーブルで、テーブルの各行はデバイス、各列は CPU を表します。テーブルの各セルには、特定のデバイスと CPU に関連する実際の割り込み数と、そのデバイスの割り込みハンドラがその CPU 上で消費した絶対時間の割合 (パーセント) が表示されます。

デバイス名は {*name*}#{*instance*} という形式で表されます。name は正規化されたドライバ名で、通常は、ドライバを実装しているモジュールの名前に対応します。ddi_driver_name(9F) を参照してください。Sun から提供されているドライバの多くには、独自のマニュアルページがあります。Intro(7) を参照してください。

標準出力が端末の場合、テーブルには端末の幅に収まるだけの列数のデータが表示されます。標準出力が端末でない場合、テーブルには最大 4 列のデータが表示されます。デフォルトでは、すべての CPU のデータが収集され、表示されます。データが 1 つのテーブルに収まらない場合は、複数のテーブルに出力されます。データを表示する CPU のセットを -c または -C オプションで指定することもできます。

デフォルトでは、intrstat は 1 秒に 1 回データを表示し、無期限に実行を続けます。どちらの動作も、それぞれ interval パラメータと count パラメータで必要に応じて制御できます。「オペランド」の項を参照してください。

intrstat では、動的発見が使用されているため、コマンドの実行中に割り込みを上げたデバイスだけが報告されます。intrstat の実行中に割り込みを上げなかったデバイスは表示されません。

intrstat を実行すると、システム全体のパフォーマンスが多少低下します。結果として、デフォルトではスーパーユーザーだけが intrstat を実行できます。『Solaris 動的トレースガイド』には、管理者がほかのユーザーに特権を付与して intrstat の実行を許可する方法が説明されています。

オプション

次のオプションがサポートされています。

-c *cpulist*
: *cpulist* で指定された CPU のデータを表示します。

  cpulist に指定できるのは、単一のプロセッサ ID (例: 4)、プロセッサ ID の範囲 (例: 4-6)、あるいはプロセッサ ID またはプロセッサ ID の範囲をコンマで区切ったリスト (例: 4,5,6 または 4,6-8) です。

-C *processor_set_id*
: *processor_set_id* で指定されたプロセッサセット内の CPU のデータを表示します。

intrstat の出力には、指定されたプロセッサセットの CPU が常に反映されます。セットに CPU が追加されると、intrstat は出力を変更してその CPU を含めます。セットから CPU が削除されると、intrstat は出力を変更してその CPU を除外します。指定できるプロセッサセットは最大 1 つです。

-T u|d
タイムスタンプを表示します。

時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。標準の日付フォーマットに d を指定します。date(1) を参照してください。

-x *opt*[=*val*]
DTrace の実行時オプションまたは D コンパイラオプションを有効化または変更します。オプションのリストは、『『Solaris Dynamic Tracing Guide』』に記載されています。ブール型のオプションを有効にするときは、その名前を指定します。値を持つオプションを設定するには、オプションの名前と値を等号 (=) で分離します

オペランド

次のオペランドがサポートされています。

count
intrstat が終了前にその出力を表示する回数を表します。

interval
intrstat 出力を表示する間隔を秒数で表します。

使用例

例 1　オプションを指定せずに intrstat を使用する

オプションを指定せずに intrstat を実行すると、トラップタイプと CPU のテーブルが表示されます。デフォルトの端末幅に収まるのは最大 4 列です。5 つ以上の CPU がある場合は、複数のテーブルが表示されます。

単一プロセッサ Intel IA/32 ベースのノートパソコンで intrstat を実行する例を次に示します。

```
example# intrstat
               device |      cpu0 %tim
     -----------------+---------------
                ata#0 |       166  0.4
                ata#1 |         0  0.0
          audioi810#0 |         6  0.0
              i8042#0 |       281  0.7
               iprb#0 |         6  0.0
               uhci#1 |         6  0.0
               uhci#2 |         6  0.0

               device |      cpu0 %tim
     -----------------+---------------
                ata#0 |       161  0.5
```

例 1　オプションを指定せずに intrstat を使用する　　(続き)

```
                ata#1 |         0  0.0
          audioi810#0 |         6  0.0
              i8042#0 |       303  0.6
               iprb#0 |         6  0.0
               uhci#1 |         6  0.0
               uhci#2 |         6  0.0
     ...
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/dtrace
インタフェースの安定性	下記を参照。

コマンド行の構文は「確実」です。人間が読める形式の出力は「不確実」です。

関連項目

dtrace(1M), trapstat(1M), attributes(5), Intro(7), ddi_driver_name(9F)

『Solaris 動的トレースガイド』

名前

in.uucpd, uucpd – UUCP server

形式

```
/usr/sbin/in.uucpd [-n]
```

機能説明

in.uucpd is the server for supporting UUCP connections over networks.

in.uucpd is invoked by inetd(1M) when a UUCP connection is established, that is, a connection to the port indicated in the "uucp" service specification, and executes the following protocol. See services(4):

1. The server prompts with login:. The uucico(1M) process at the other end must supply a username.
2. Unless the username refers to an account without a password, the server then prompts with Password:. The uucico process at the other end must supply the password for that account.

If the username is not valid, or is valid but refers to an account that does not have /usr/lib/uucp/uucico as its login shell, or if the password is not the correct password for that account, the connection is dropped. Otherwise, uucico is run, with the user ID, group ID, group set, and home directory for that account, with the environment variables USER and LOGNAME set to the specified username, and with a -u flag specifying the username. Unless the -n flag is specified, entries are made in /var/adm/utmpx, /var/adm/wtmpx, and /var/adm/lastlog for the username. in.uucpd must be invoked by a user with appropriate privilege (usually root) in order to be able to verify that the password is correct.

セキュリティー

in.uucpd uses pam(3PAM) for authentication, account management, and session management. The PAM configuration policy, listed through /etc/pam.conf, specifies the modules to be used for in.uucpd. Here is a partial pam.conf file with entries for uucp using the UNIX authentication, account management, and session management module.

```
uucp    auth requisite          pam_authtok_get.so.1
uucp    auth required           pam_dhkeys.so.1
uucp    auth required           pam_unix_auth.so.1

uucp    account requisite       pam_roles.so.1
uucp    account required        pam_projects.so.1
uucp    account required        pam_unix_account.so.1

uucp      session required      pam_unix_session.so.1
```

If there are no entries for the uucp service, then the entries for the "other" service will be used. If multiple authentication modules are listed, then the peer may be prompted for multiple passwords.

ファイル

/var/adm/utmpx	accounting
/var/adm/wtmpx	accounting
/var/adm/lastlog	time of last login

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

svcs(1), inetadm(1M), inetd(1M), svcadm(1M), uucico(1M), pam(3PAM), pam.conf(4), services(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5)

診断

All diagnostic messages are returned on the connection, after which the connection is closed.

`user read` An error occurred while reading the username.

`passwd read` An error occurred while reading the password.

`Login incorrect.` The username is invalid or refers to an account with a login shell other than `/usr/lib/uucp/uucico`, or the password is not the correct password for the account.

注意事項

The `in.uucpd` service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/uucp
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

iostat – 入出力統計を報告する

形式

```
/usr/bin/iostat  [-cCdDeEiImMnpPrstxXYz] [-l n] [-T u | d]
 [disk]... [interval [count]]
```

機能説明

iostat ユーティリティーはターミナル、ディスク、テープの I/O 活動、CPU 使用率を繰り返し報告します。出力の最初の行はブートから全時間の統計です;続く各行はその前の間隔からのみの統計です。

この情報を計算するために、カーネルは多数のカウンタを保持しています。各ディスクに対し、カーネルは読み取り、書き込み、読み取りバイト数、書き込みバイト数を数えます。また、カーネルは、キューの入り口と出口で高解像度のタイムスタンプを使用します。これによって、滞留時間や、滞留時間と長さの積の累積をキューごとに記録することができます。これらの値を使って、iostat はスループット、使用率、キュー長、トランザクション率、サービス時間を非常に正確に計測します。端末全般については、カーネルは単純に入出力の文字数をカウントします。

カーネルステータスコマンドの実行中、システムの state は変わる場合があります。もし関連があれば、状態が変わったメッセージは以下の形式のどれかで iostat の出力に含まれます:

```
<<device added: sd0>>
<<device removed: sd0>>
<<partition added: sd0,a>>
<<partition removed: sd0,a>>
<<NFS mounted: nfs1>>
<<NFS unmounted: nfs1>>
<<multi-path added: ssd4>>
<<multi-path removed: ssd4>>
<<controller added: c1>>
<<controller removed: c1>>
<<processors added: 1, 3>>
<<processors removed: 1, 3>>
```

これらの状態変化メッセージに表示される names は -n や -m オプションによって適切なものに変えられます。

より全般的なシステム統計情報を参照するには、sar(1)、sar(1M)、または vmstat(1M) を使ってください。

出力

iostat ユーティリティーの出力は次の情報を含みます。

device	ディスクの名前
r/s	秒あたり読み込み数
w/s	秒あたり書き込み数

`kr/s` 秒あたり読み込みキロバイト数

インターバルの平均 I/O サイズは、`kr/s` を `r/s` で割って求めることができます。

`kw/s` 秒あたり書き込みキロバイト数

インターバルの平均 I/O サイズは、`kw/s` を `w/s` で割って求めることができます。

`wait` サービスを待っている平均トランザクション数 (キュー長)

これは、デバイスによる受け入れを待っている、デバイスドライバキューに保持されている I/O 操作の数です。

`actv` 現在サービスを受けている (キューからは削除されたがまだ完了していない) 平均トランザクション数

これは、デバイスによって受け入れられたがまだサービスが終わっていない I/O 操作の数です。

`svc_t` トランザクションの平均レスポンス時間 (ミリ秒単位)

`svc_t` 出力はデバイスの *service* 時間よりはむしろ全体的な *response* 時間を報告します。全体的な時間はトランザクションがキューにいた時間とサービスを受けた時間を含みます。キューで過ごした時間は `-x` オプションによって `wsvc_t` 出力列に表示されます。サービストランザクションに費やされた時間は本当の実行時間です。サービス時間もまた `-x` オプションによって同じレポートの `asvc_t` 出力列に表示されます。

`%w` サービスを待っていたトランザクションがあった (キューが空でない) 時間の割合

`%b` ディスクがビジー状態にある (トランザクションが進行中である) 時間の割合

`wsvc_t` 待ち行列 (キュー) での平均サービス時間 (ミリ秒単位)

`asvc_t` アクティブなトランザクションの平均サービス時間 (ミリ秒単位)

`wt` I/O 待ち時間は CPU 時間の割合としては計算されず、この統計は常にゼロを返します。

オプション

次のオプションがサポートされています。

`-c` システムがユーザーモード、システムモード、 I/O 待ち状態、アイドル状態のそれぞれに費やした時間の割合を報告します。詳細な情報については「注意事項」項目をご覧ください。

-C　-x オプションも指定された場合は、 *controller id* によって集計された拡張ディスク統計を報告します。

-d　各ディスクに関し、秒あたり転送キロバイト数、秒あたり転送数、ミリ秒あたりサービス時間を報告します。

-D　各ディスクに関し、秒あたり転送キロバイト数、秒あたり転送数、およびサービス時間 (ミリ秒単位) を報告します。

-e　デバイスエラー要約統計情報を表示します。合計エラー数、ハードエラー数、ソフトエラー数、転送エラー数が表示されます。

-E　全デバイスエラー統計情報を表示します。

-i　-E 出力で、 Serial No の代わりに Device ID を表示します。Device Id はデバイスドライバーが ddi_devid_register(9F) を使用して登録した固有の識別子です。

-I　各間隔で、(適切な場合は) 率ではなくカウントを報告します。

-l *n*　報告の中でディスクの数を *n* に制限します。デフォルトでは -d と -D のときにはディスク数を 4 に制限し、 -x では制限しません。注意: 個別に指定したディスク (以下の *disk* 参照) には、このディスク制限は適用されません。

-m　ファイルシステムのマウントポイントを報告します。このオプションは -P か -p オプションと共に指定した場合あるいは -Xn か -en と共に使用された場合にもっとも役立ちます。 -m オプションはマウントポイントが実際に出力に現れた場合にのみ役立ちます。このオプションは　-n オプションと一緒にのみ使用できます。

-M　データスループットを KB/sec の代わりに MB/sec で表示します。

-n　名前を説明的な形式で表示します。たとえば、cXtYdZ、rmt/N、server:/export/path などです。

デフォルトでは、ディスクは ssd23 や md301 のようなインスタンス名で識別されます。-n オプションを -x オプションと組み合わせることで、ディスク名を cXtYdZsN 形式で表示することができます。こうすることで、ディスク名を物理的ハードウェア特性により関連づけやすくなります。cXtYdZsN 形式は、特にファイバーチャネル (FC) 環境で FC ワールドワイドネームが t フィールドに表示されるため便利です。

-p　各ディスクで、デバイスごとの統計に加え、パーティションごとの統計を報告します。

-P　各ディスクで、パーティションごとの統計のみを報告し、デバイスごとの統計は報告しません。

-r　コンマ区切り形式でデータを表示します。

-s　state changes に関するメッセージを抑止します。

-t　秒あたりの読み取りと端末への書き込み文字数を報告します。

-T u|d　タイムスタンプを表示します。

時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。 標準の日付フォーマットに d を指定します。date(1) を参照してください。

-X　scsi_vhci(7D) の管理下にあるディスクについては、ディスクの *lun* 統計情報に付け加え、*lun .controller* の統計情報も報告します。

-x　拡張ディスク統計情報を報告します。デフォルトでは、ディスクは ssd23 や md301 のようなインスタンス名で識別されます。x オプションを -n オプションと組み合わせることで、ディスク名を cXtYdZsN 形式で表示することができます。こうすることで、ディスク名を物理的ハードウェア特性により関連づけやすくなります。cXtYdZsN 形式は、特にファイバーチャネル (FC) 環境で FC ワールドワイドネームが t フィールドに表示されるため便利です。

出力表示が要求されない場合は、(-x、-e、-E が指定されない場合は)、-x が適用されます。

-Y　scsi_vhci(7D) の管理下にあるディスクについては、ディスクの *lun* 統計情報に付け加え、*lun.targetport* と *lun.targetport.controller* の統計情報も報告します。

-n (説明的) モードではパスの *target-port* プロパティーを使用して targetport が表示されます。-n が指定されない場合は、より短い port-id を使用して *targetport* が表示されます。同じ target-port プロパティー値を持つターゲットポートは、すべて同じ port-id を共有します。target-port と port-id の関連付けはリブート後にはなくなります。

出力表示が要求されない場合は、(-x、-e、-E が指定されない場合は)、-x が適用されます。

-z　その行のデータがすべてゼロの場合には行を出力しません。

-xcnCXTdz *interval* オプションセットは特にディスク I/O 問題が存在するかどうかを判断したり、問題を特定したりする場合に役立ちます。

オペランド

次のオペランドがサポートされています。

count　*count* 回のみ報告します。

disk　報告する対象ディスクを個別に指定します。指定されたディスクに加えて、ディスク数の上限に達するまでは (上記の -l 参照)、アクティブなディスクも報告されます。

interval　各 *interval* 秒ごとに報告します。

使用例

例 1　iostat を使用してユーザーおよびシステムオペレーション統計情報を生成する

次のコマンドは *controller id* によって集約された、ユーザー (us) 操作とシステム (sy) 操作に関する 2 つの拡張デバイス統計報告を表示します。-n オプションは -x オプションとともに使用されるため、デバイスはコントローラ名で識別されます。

```
example% iostat -xcnCXTdz 5


Mon Nov 24 14:58:36 2003
    cpu
 us sy wt id
 14 31  0 20
                    extended device statistics
  r/s    w/s    kr/s      kw wait  actv wsvc_t asvc_t  %w  %b device
  3.8   29.9   145.8    44.0  0.0   0.2    0.1    6.4   0   5     c0
666.3  814.8 12577.6 17591.1 91.3  82.3   61.6   55.6   0   2    c12
180.0  234.6  4401.1  5712.6  0.0 147.7    0.0  356.3   0  98    d10

Mon Nov 24 14:58:41 2003
    cpu
 us sy wt id
 11 31  0 22
                    extended device statistics
  r/s    w/s    kr/s      kw wait  actv wsvc_t asvc_t  %w  %b device
  0.8   41.0     5.2    20.5 0.0    0.2    0.2    4.4   0   6     c0
565.3  581.7  8573.2 10458.9 0.0   26.6    0.0   23.2   0   3    c12
106.5   81.3  3393.2  1948.6 0.0    5.7    0.0   30.1   0  99    d10
```

例 2　iostat を使用して TTY 統計を生成する

次のコマンドは、異なるオペレーションモードの 5 つのディスク活動に関して、2 つの報告を表示します。-x オプションが使用されているため、ディスクはインスタンス名で識別されています。

```
example% iostat -x tc 5 2


                  extended device statistics        tty         cpu
device r/s  w/s kr/s  kw/s wait actv svc_t %w  %b  tin tout  us sy wt id
sd0    0.4  0.3 10.4   8.0  0.0  0.0  36.9  0   1    0   10   0  0  0 99
sd1    0.0  0.0  0.3   0.4  0.0  0.0  35.0  0   0
sd6    0.0  0.0  0.0   0.0  0.0  0.0   0.0  0   0
nfs1   0.0  0.0  0.0   0.0  0.0  0.0   0.0  0   0
nfs2   0.0  0.0  0.0   0.1  0.0  0.0  35.6  0   0
            extended device statistics              tty         cpu
device r/s  w/s  kr/s  kw/s wait actv svc_t %w  %b tin tout  us sy wt id
```

例 2　iostat を使用して TTY 統計を生成する　　*(続き)*

```
sd0    0.0  0.0  0.0   0.0  0.0  0.0  0.0   0   0   0  155   0  0  0 100
sd1    0.0  0.0  0.0   0.0  0.0  0.0  0.0   0   0
sd6    0.0  0.0  0.0   0.0  0.0  0.0  0.0   0   0
nfs1   0.0  0.0  0.0   0.0  0.0  0.0  0.0   0   0
nfs2   0.0  0.0  0.0   0.0  0.0  0.0  0.0   0   0
```

例 3　iostat を使用してパーティションとデバイスの統計を生成します。

次のコマンドは各ディスクのパーティションとデバイス統計を生成します。-n オプションは -x オプションとともに使用されているため、ディスクはコントローラ名で識別されています。

```
example% iostat -xnp

                 extended device statistics
r/s  w/s  kr/s kw/s wait actv wsvc_t asvc_t %w %b device
0.4  0.3  10.4  7.9  0.0  0.0    0.0   36.9  0  1 c0t0d0
0.3  0.3   9.0  7.3  0.0  0.0    0.0   37.2  0  1 c0t0d0s0
0.0  0.0   0.1  0.5  0.0  0.0    0.0   34.0  0  0 c0t0d0s1
0.0  0.0   0.0  0.1  0.0  0.0    0.6   35.0  0  0 fuji:/export/home/user3
```

例 4　インスタンス名から説明的名前へ翻訳する

次の例は iostat によって特定のインスタンス名を説明的名前に翻訳します。

```
example% iostat -xn sd1
                        extended device statistics
r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c8t1d0
```

例 5　特定のディスクのターゲットポートとコントローラアクティビティーを表示する

次の例では、同じターゲットポートに接続された 4 つのコントローラがあります。

```
# iostat -Y ssd22
                    extended device statistics
device          r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b
ssd22           0.2    0.0    1.5    0.0  0.0  0.0    0.7   0   0
ssd22.t2        0.2    0.0    1.5    0.0  0.0  0.0    0.0   0   0
ssd22.t2.fp0    0.0    0.0    0.4    0.0  0.0  0.0    0.0   0   0
ssd22.t2.fp1    0.0    0.0    0.4    0.0  0.0  0.0    0.0   0   0
ssd22.t2.fp2    0.0    0.0    0.4    0.0  0.0  0.0    0.0   0   0
ssd22.t2.fp3    0.0    0.0    0.4    0.0  0.0  0.0    0.0   0   0
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照

この呼び出しは開発中です。人間が読める形式の出力は不安定 (Unstable) です。

関連項目

date(1), sar(1), sar(1M), mpstat(1M), vmstat(1M), time(2), attributes(5), scsi_vhci(7D)

注意事項

CPU 使用率の合計は製品の百分率の数字中の丸め誤差のために 100 と若干異なるかもしれません。

svc_t 応答時間は特に I/0 (r/s+w/s) 率が毎秒 0.5 以下の場合はっきりしません。このような場合、問題のない数値の急上昇がよく見られます。

mpstat ユーティリティーは同じ wt、usr、sys 統計情報を報告します。詳細については、mpstat(1M) を参照してください。

zone で実行された場合、かつ、プールファシリティーがアクティブだった場合は、iostat(1M) は zone が割り当てられているプールのプロセッサセットの中のプロセッサの情報のみを表示します。

名前

ipaddrsel – configure IPv6 default address selection

形式

```
/usr/sbin/ipaddrsel
/usr/sbin/ipaddrsel -f file
/usr/sbin/ipaddrsel -d
```

機能説明

Use the `ipaddrsel` utility to configure the IPv6 default address selection policy table. The policy table is a longest-matching-prefix lookup table that is used for IPv6 source address selection and for destination address ordering when resolving names to `AF_INET6` addresses. For a description of how the policy table is used for source address selection, see `inet6`(7P). For a description of how the policy table is used for destination address ordering, see `getaddrinfo`(3SOCKET).

The unmodified policy table is valid for all typical IPv6 deployments. Modify the table only if a circumstance exists for which the default behavior of the IPv6 source address selection or destination address ordering mechanism is unsatisfactory. See the 使用例 section for examples of such circumstances. You should carefully consider your addressing strategy before you change the table from the provided default.

When the `ipaddrsel` command is issued without any arguments, the address selection policy currently in use is printed. The format of the output is compatible with the format of the configuration file that the `-f` option accepts.

注 – If the `usesrc` subcommand to `ifconfig`(1M) is applied to a particular physical interface, the selection policy specified by `usesrc` overrides the source address selection policies specified by `ipaddrsel`. This is true for packets that are locally generated and for applications that do not choose a non-zero source address using `bind`(3SOCKET).

The Configuration File

The configuration file that the `-f` option accepts can contain either comment lines or policy entries. Comment lines have a '#' character as the first non-blank character. and they are ignored by the `ipaddrsel` utility. Policy entry lines have the following format:

prefix/prefix_length precedence label [# *comment*]

The *prefix* must be an IPv6 prefix in a format consistent with `inet`(3SOCKET). The *prefix_length* is an integer ranging from 0 to 128. The IPv6 source address selection and destination address ordering algorithms determine the precedence or label of an address by doing a longest-prefix-match lookup using the prefixes in this table, much like next-hop determination for a destination is done by doing a longest-prefix-match lookup using an IP routing table.

The precedence is a non-negative integer that represents how the destination address ordering mechanism will sort addresses returned from name lookups. In general, addresses with a higher precedence will be in front of addresses with a lower precedence. Other factors, such as destinations with undesirable source addresses can, however, override these precedence values.

The label is a string of at most fifteen characters, not including the `NULL` terminator. The label allows particular source address prefixes to be used with destination prefixes of the same label. Specifically, for a particular destination address, the IPv6 source address selection algorithm prefers source addresses whose label is equal that of the destination.

The label may be followed by an optional comment.

The file must contain a default policy entry, which is an entry with `::0/0` as its *prefix* and *prefix_length*. This is to ensure that all possible addresses match a policy.

オプション

The `ippadrsel` utility supports the following options:

`-f` *file* Replace the address selection policy table with the policy specified in the *file*.

`-d` Revert the kernel's address selection policy table back to the default table. Invoking `ipaddrsel` in this way only changes the currently running kernel's policy table, and does not alter the configuration file `/etc/inet/ipaddrsel.conf`. To revert the configuration file back to its default settings, use `ipaddrsel -d`, then dump the contents of the table to the configuration file by redirecting the output of `ipaddrsel` to `/etc/inet/ipaddrsel.conf`.

```
example# ipaddrsel -d
example# ipaddrsel > /etc/inet/ipaddrsel.conf
```

使用例

例 1 The Default Policy in /etc/inet/ipaddrsel.conf

The following example is the default policy that is located in `/etc/inet/ipaddrsel.conf`:

```
# Prefix                          Precedence Label
::1/128                           50 Loopback
::/96                             20 IPv4_Compatible
::ffff:0.0.0.0/96                 10 IPv4
2002::/16                         30 6to4
::/0                              40 Default
```

例 2 Assigning a Lower Precedence to Link-local and Site-local Addresses

By default, the destination address ordering rules sort addresses of smaller scope before those of larger scope. For example, if a name resolves to a global and a site-local address, the site local address would be ordered before the global address. An administrator can override this ordering rule by assigning a lower precedence to addresses of smaller scope, as the following table demonstrates.

```
# Prefix                          Precedence Label
::1/128                           50 Loopback
::/0                              40 Default
2002::/16                         30 6to4
fec0::/10                         27 Site-Local
fe80::/10                         23 Link-Local
::/96                             20 IPv4_Compatible
```

例 2 Assigning a Lower Precedence to Link-local and Site-local Addresses *(続き)*

```
::ffff:0.0.0.0/96                    10 IPv4
```

例 3 Assigning Higher Precedence to IPv4 Destinations

By default, IPv6 addresses are ordered in front of IPv4 addresses in name lookups. `::ffff:0.0.0.0/96` has the lowest precedence in the default table. In the following example, IPv4 addresses are assigned higher precedence and are ordered in front of IPv6 destinations:

```
# Prefix                             Precedence Label
::1/128                              50 Loopback
::/0                                 40 Default
2002::/16                            30 6to4
::/96                                20 IPv4_Compatible
::ffff:0.0.0.0/96                    60 IPv4
```

例 4 Ensuring that a Particular Source Address is Used

This example ensures that a particular source address is used only when communicating with destinations in a particular network.

The following policy table assigns a label of 5 to a particular source address on the local system, `2001:1111:1111::1`. The table assigns the same label to a network, `2001:2222:2222::/48`. The result of this policy is that the `2001:1111:1111::1` source address will only be used when communicating with destinations contained in the `2001:2222:2222::/48` network. For this example, this network is the `ClientNet`, which could represent a particular client's network.

```
# Prefix                             Precedence Label
::1/128                              50 Loopback
2001:1111:1111::1/128                40 ClientNet
2001:2222:2222::/48                  40 ClientNet
::/0                                 40 Default
2002::/16                            30 6to4
::/96                                20 IPv4_Compatible
::ffff:0.0.0.0/96                    10 IPv4
```

This example assumes that the local system has one physical interface, and that all global prefixes are assigned to that physical interface.

終了ステータス

`ipaddrsel` returns the following exit values:

`0` `ipaddrsel` successfully completed.

`>0` An error occurred. If a failure is encountered, the kernel's current policy table is unchanged.

ファイル

`/etc/inet/ipaddrsel.conf` The file that contains the IPv6 default address selection policy to be installed at boot time. This file is loaded before any Internet services are started.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`nscd(1M)`, `inet(3SOCKET)`, `getaddrinfo(3SOCKET)`, `ipaddrsel.conf(4)`, `attributes(5)`, `inet6(7P)`

注意事項

The ipnodes cache kept by `nscd(1M)` contains addresses that are ordered using the destination address ordering algorithm, which is one of the reasons why `ipaddrsel` is called before `nscd` in the boot sequence. If `ipaddrsel` is used to change the address selection policy after `nscd` has started, you should invalidate the `nscd` ipnodes cache invalidated by invoking the following command:

```
example# /usr/sbin/nscd -i ipnodes
```

名前

ipadm – インターネットプロトコルネットワークインタフェースおよび TCP/IP 調整可能パラメータの構成

形式

```
ipadm create-ip [-t] IP-interface
ipadm delete-ip IP-interface

ipadm create-vni [-t] VNI-interface
ipadm delete-vni VNI-interface

ipadm create-ipmp [-t] [-i interface,[...]...] IPMP-interface
ipadm delete-ipmp [-f] IPMP-interface
ipadm add-ipmp [-t] -i interface,[...] [-i interface,[...]...]
     IPMP-interface
ipadm remove-ipmp [-t] -i interface,[...] [-i interface,[...]...]
     IPMP-interface

ipadm show-if [[-p] -o field[,...]] [interface]
ipadm disable-if -t interface
ipadm enable-if -t interface

ipadm set-ifprop [-t] -m protocol -p prop=value[,...] interface
ipadm reset-ifprop [-t] -m protocol -p prop interface
ipadm show-ifprop [[-c] -o field[,...]] [-p prop,...]
     [-m protocol] [interface]

ipadm create-addr [-t] -T static [-d]
     -a {local|remote}=addr[/prefixlen],... addrobj
ipadm create-addr [-t] -T dhcp [-w seconds | forever]
     [-h hostname] addrobj
ipadm create-addr [-t] -T addrconf [-i {local|remote}=interface-id]
     [-p {stateful|stateless}={yes|no},..] addrobj

ipadm delete-addr [-r] addrobj
ipadm show-addr [[-p] -o field[,...]] [-d] [addrobj | interface/]
ipadm up-addr [-t] addrobj
ipadm down-addr [-t] addrobj
ipadm refresh-addr [-i] addrobj
ipadm disable-addr -t addrobj
ipadm enable-addr -t addrobj

ipadm set-addrprop [-t] -p prop=value[,...] addrobj
ipadm reset-addrprop [-t] -p prop=value[,...] addrobj
ipadm show-addrprop [[-c] -o field[,...]] [-p prop[,...]] [addrobj]

ipadm set-prop [-t] -p prop=value[,...] protocol
ipadm reset-prop [-t] -p prop protocol
ipadm show-prop [[-c] -o field[,...]] [-p prop[,...] protocol | protocol]
```

機能説明

`ipadm` コマンドには一連のサブコマンドがあり、次の目的に使用できます。

インタフェースを管理する:

- インタフェースクラス `ip`、`ipmp`、および `vni` のインタフェースを作成および削除する

- インタフェースのプロパティーを変更する
- インタフェースの構成を表示する

アドレスを管理する:

- アドレスを作成および削除する
- アドレスのプロパティーを変更する
- アドレスの構成を表示する

TCP/IP プロトコルのプロパティーを管理する:

- TCP/IP のプロパティーを変更する
- TCP/IP のプロパティーを表示する

ipadm サブコマンドのさまざまなオペランドについては、「サブコマンド」に続く「オペランド」の項で説明されています。

必要な承認と特権

次のサブコマンドには、solaris.network.interface.config 承認と PRIV_SYS_IP_CONFIG 特権が必要です。

```
create-ip          create-addr
delete-ip          up-addr
create-vni         down-addr
delete-vni         refresh-addr
create-ipmp        disable-addr
delete-ipmp        enable-addr
add-ipmp           set-addrprop
remove-ipmp        reset-addrprop
disable-if         set-prop
enable-if          reset-prop
set-ifprop
reset-ifprop
```

前述の承認と特権に加えて、ipadm のサブコマンドの create-ip、create-vni、create-ipmp、および enable-if には PRIV_NET_RAWACCESS 特権も必要です。

サブコマンド

次のサブコマンドがサポートされています。

create-ip [-t] *IP-interface*

IPv4 パケットと IPv6 パケットの両方を処理する IP インタフェースを作成します。IPv4 インタフェースのアドレスは 0.0.0.0 に設定され、IPv6 インタフェースのアドレスは :: に設定されます。デフォルトでは、このサブコマンドの情報は持続し、次回のリブート時にこのインタフェースがインスタンス化されます。

インタフェースの作成時には、IPv4 と IPv6 の両方が暗黙的に有効になっています。インタフェースの無効化および有効化については、後述の disable-if サブコマンドと enable-if サブコマンドを参照してください。

lo0 は、ループバックインタフェースと呼ばれる特殊なインタフェースです。これは仮想 IP インタフェースなので、どの物理ハードウェアにも関連付けられません。これはシステムに最初に作成される IP インタフェースの 1 つであり、その IPv4 アドレスは 127.0.0.1、IPv6 アドレスは ::/128 です。

-t、--temporary
これは一時的な操作で、永続してはいけないことを示します。操作の影響を受けるのはアクティブな構成だけです。

delete-ip *IP-interface*
この IP インタフェースをアクティブな構成から削除します。このインタフェースに構成されているすべてのアドレスが破棄されます。さらに、インタフェースに関連するすべての永続的な情報が永続的データストアから削除されるため、リブート時に *interface* はインスタンス化されません。アクティブな構成からインタフェースを (削除するのではなく) 無効にするには、disable-if サブコマンドを使用します。

create-vni [-t] *VNI-interface*
IPv4 パケットと IPv6 パケットの両方を処理する VNI インタフェースを作成します。IPv4 インタフェースのアドレスは 0.0.0.0 に設定され、IPv6 インタフェースのアドレスは :: に設定されます。デフォルトでは、このサブコマンドの情報は持続し、次回のリブート時にこのインタフェースがインスタンス化されます。

インタフェースの作成時には、IPv4 と IPv6 の両方が暗黙的に有効になっています。インタフェースの無効化および有効化については、後述の disable-if サブコマンドと enable-if サブコマンドを参照してください。

仮想インタフェースなのでどのハードウェアにも関連付けられないという点で、vni は特殊なインタフェースです。vni(7d) を参照してください。

-t, -—temporary
これは一時的な操作で、持続してはいけないことを示します。操作の影響を受けるのはアクティブな構成だけです。

delete-vni *VNI-interface*
この VNI インタフェースをアクティブな構成から削除します。このインタフェースに構成されているすべてのアドレスが破棄されます。さらに、IP インタフェースに関連するすべての永続的な情報が永続的データストアから削除されるため、リブート時に interface はインスタンス化されません。アクティブな構成からインタフェースを (削除するのではなく) 無効にするには、disable-if サブコマンドを使用します。

create-ipmp [-t] [-i *interface*,[...]...] *IPMP-interface*
IPv4 パケットと IPv6 パケットの両方を処理する IPMP インタフェースを作成します。IPv4 インタフェースのアドレスは 0.0.0.0 に設定され、IPv6 インタフェースのアドレスは :: に設定されます。デフォルトでは、このサブコマンドの情報は持続し、次回のリブート時にこのインタフェースがインスタンス化されます。

インタフェースの作成時には、IPv4 と IPv6 の両方が暗黙的に有効になっています。IPMP インタフェースの無効化および有効化については、後述の `disable-if` サブコマンドと `enable-if` サブコマンドを参照してください。

`-t`, `--temporary`
これは一時的な操作で、持続してはいけないことを示します。操作の影響を受けるのはアクティブな構成だけです。

`-i`, `--interface` *interface*,[...]
基になるインタフェースとして IPMP インタフェースに追加するインタフェースのコンマ区切りのリスト。インタフェースを IPMP グループに正常に追加するには、これらのインタフェースはアクティブな構成に存在している必要があり、ほかの IPMP グループに属していてはいけません。`-i` オプションは複数指定できます。IPMP インタフェースは作成されたが、指定された基になるインタフェースが 1 つも正常に追加されなかった場合、このコマンドは部分的な成功を返します。

`delete-ipmp` [`-f`] *IPMP-interface*
この IPMP インタフェースをアクティブな構成から削除します。このインタフェースに構成されているすべてのアドレスが破棄されます。IPMP インタフェースに基になるインタフェースが存在している場合、`-f` オプションを指定しないと、このコマンドは失敗します。さらに、IPMP インタフェースに関連するすべての永続的な情報が永続的データストアから削除されるため、リブート時に interface はインスタンス化されません。アクティブな構成からインタフェースを (削除するのではなく) 単に無効にするには、`disable-if` サブコマンドを使用します。

`-f`, `--force`
IPMP インタフェースに基になるインタフェースが存在している場合、このオプションを指定すると、基になるインタフェースがすべてグループから削除されたあと、IPMP インタフェースが削除されます。

`add-ipmp` [`-t`] `-i` *interface*,[...] [`-i` *interface*,[...]...] *IPMP-interface*
指定された IPMP インタフェースに 1 つまたは複数の基になる IP インタフェースを追加します。

`-t`, `--temporary`
これは一時的な操作で、持続してはいけないことを示します。操作の影響を受けるのはアクティブな構成だけです。

`-i`, `--interface` *interface*,[...]
基になるインタフェースとして IPMP インタフェースに追加するインタフェースのコンマ区切りのリスト。インタフェースを IPMP グループに正常に追加するには、これらのインタフェースはアクティブな構成に存在している必要があり、ほかの IPMP グループに属していてはいけません。少なくとも 1 つのインタフェースは追加されたが、残りのインタフェースは追加に失敗した場合、このコマンドは部分的な成功を返します。`-i` オプションは複数指定できます。

`remove-ipmp` [`-t`] `-i` *interface*,[...] [`-i` *interface*,[...]...] *IPMP-interface*

この IPMP インタフェースから 1 つまたは複数の基になる IP インタフェースを削除します。

`-t`, `--temporary`

これは一時的な操作で、持続してはいけないことを示します。操作の影響を受けるのはアクティブな構成だけです。

`-i`, `--interface` *interface*,[...]

IPMP インタフェースから削除する基になるインタフェースのコンマ区切りのリスト。指定するインタフェースは、この IPMP グループの基になるインタフェースとしてすでに設定されている必要があります。`-i` オプションは複数指定できます。少なくとも 1 つのインタフェースは削除されたが、残りのインタフェースは削除に失敗した場合、このコマンドは部分的な成功を返します。

`show-if` [[`-p`] `-o` *field*[,...]] [*interface*]

ネットワークインタフェースの構成情報を表示します。システム上に構成されているすべてのネットワークインタフェース (永続的な構成だけに存在するものも含む) を対象にすることも、特定のネットワークインタフェースを対象にすることもできます。

`-o` *field*[,...], `--output` *field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。*field* 名には、次に示すフィールドのいずれか、あるいは、すべてのフィールドを表示するための特殊な値 `all` を指定する必要があります。各ネットワークインタフェースについて、次のフィールドを表示できます。

`IFNAME`

IP インタフェースの名前。

`CLASS`

次のうちのいずれかを指定します。

`ip`

基になるデータリンク上に plumb されたインタフェース。

`ipmp`

1 つまたは複数の基になる IP インタフェース上に作成された IPMP インタフェース。

`loopback`

ループバックインタフェース。

`vni`

仮想 IP インタフェース。`vni(7d)` を参照してください。

`STATE`

表示されるインタフェースに関して、次のいずれかを示します。

ok
インタフェースが必要とするリソースが確保されていることを示します。一部のインタフェースでは、これは、リンクが稼働していることも示します。

offline
インタフェースはオフライン状態なので、IP データトラフィックを送受信できません。if_mpadm(1M) を参照してください。

failed
データリンクが停止していることを示します。インタフェースが IPMP グループに属している場合、これは、インタフェースに障害が発生している (つまり、IFF_FAILED が設定されている) ことも示すことがあります。障害が発生したインタフェースは、IP データトラフィックの送受信には使用されません。これが IPMP グループの物理 IP インタフェースに設定されている場合、IP データトラフィックは、その IPMP グループで使用可能なほかの IP インタフェースを介して引き続き送受信されます。これが IPMP IP インタフェースに設定されている場合は、グループ全体が障害状態であり、そのグループのどのインタフェースでもデータトラフィックを送受信できません。in.ndpd(1M) を参照してください。

down
そのインタフェースを介して IP パケットが送受信されないように、インタフェースが管理上停止されていることを示します。

disabled
インタフェースが disable-if サブコマンドによってアクティブな構成から無効にされていることを示します。

ACTIVE
その IP インタフェースがシステムで IP データトラフィックに使用されているかどうかによって、yes または no になります。

CURRENT
アクティブな構成のインタフェースオブジェクトに関して、次のフラグのいずれかを示します。

b
インタフェースはブロードキャストをサポートしています

m
インタフェースはマルチキャストをサポートしています

p
インタフェースはポイントツーポイントリンクです

v
インタフェースは仮想インタフェースです (vni(7d)、ループバックなど)。つまり、物理インタフェースには基になるハードウェアがありません。

s
IPMP インタフェースは、管理上 standby としてマークされています。in.ndpd(1M) を参照してください。

l
インタフェースは、IPMP インタフェースの基になるインタフェースです。in.ndpd(1M) を参照してください。

i
基になるインタフェースはアクティブではありません。in.ndpd(1M) を参照してください。

V
インタフェースは VRRP インタフェースです

a
VRRP インタフェースはアクセプトモードです (~IFF_NOACCEPT)

Z
インタフェースの IP アドレスのレイヤー 3 保護が管理上適用されています。

4
インタフェースは IPv4 パケットを処理できます

6
インタフェースは IPv6 パケットを処理できます

b と p を同時に指定することはできません。

PERSISTENT
インタフェースオブジェクトがリブート時にインスタンス化されるとき、または enable-if サブコマンドで再度有効化されるときに適用する構成を指定します。s、l、4、および 6 の任意の組み合わせにできます (前述の説明を参照)。このフィールドはデフォルトでは表示されず、-o で all または persistent を指定した場合のみ表示されます。

OVER
IPMP インタフェースが作成される基になるインタフェース。これは、ほかのインタフェースクラスには適用されません。

-p, --parsable
安定したマシン解析可能な形式を使用して表示します。このオプションには -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

disable-if -t *interface*
指定された *interface* をアクティブな構成から削除することによって無効にします。このインタフェースに構成されているすべてのアドレスが無効になります。もともと永続的なインタフェースオブジェクトとして作成されていた場

合、永続的な構成は変更されません。このインタフェースを再度有効にするには、`enable-if` を使用するようにしてください。

`-t`, `--temporary`
この無効化は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`enable-if -t` *interface*
構成を永続的ストアから読み取ることで、指定されたインタフェースを有効にします。永続的なインタフェースプロパティーがあればすべて適用され、指定されたインタフェースに永続的アドレスがあればすべて有効になります。

`-t`, `--temporary`
この有効化は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`set-ifprop` [`-t`] `-m` *protocol* `-p` prop=*value*[,...] *interface*
インタフェースのプロパティーを、ユーザーが指定した値に変更します。プロパティーが複数の値を取る場合は、区切り文字としてコンマを使用して値を指定します。一度に指定できるプロパティーは 1 つだけです。インタフェースでサポートされているプロパティーおよびその指定可能な値は、`show-ifprop` サブコマンドで取得できます。一度に変更できるプロパティーは 1 つだけです。

`-t`, `--temporary`
この変更は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`-m` *protocol*, `--module` *protocol*
プロパティーを IPv4 パケットに適用するか IPv6 パケットに適用するかを指定します。

`-p` prop=*value*[,...], `-prop` prop=*value*[,...]
指定の値に設定するプロパティー。

`reset-ifprop` [`-t`] `-m` *protocol* `-p` *prop interface*
指定されたインタフェースのプロパティーをデフォルト値にリセットします。`-t` を省略すると、プロパティーの持続的な値はすべて削除されます。一度に変更できるプロパティーは 1 つだけです。

`-t`, `--temporary`
このリセットは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`-m` *protocol*, `--module` *protocol*
リセットするプロパティーを IPv4 パケットに適用するか IPv6 パケットに適用するかを指定します。

`-p` *prop*, `-prop` *prop*
指定の値に設定するプロパティー。

`show-ifprop` [[`-c`] `-o` *field*[,...]] [`-p` *prop*,...] [`-m` *protocol*] [*interface*]

作成されているすべてのインタフェースまたは特定のインタフェースについて、1 つまたは複数のプロパティーの現在の値と永続的な値を表示します。一度に複数のプロパティーを取得するには、プロパティー名をコンマで区切って `-p` オプションに指定します。`-p` オプションを指定しない場合、 使用可能なすべてのインタフェースプロパティーが表示されます。

`-o` *field*[,...], `--output` *field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。*field* 名には、次に示すフィールドのいずれか、あるいは、すべてのフィールドを表示するための特殊な値 `all` を指定する必要があります。インタフェースごとに、次のフィールドを表示できます。

`IFNAME`

インタフェースの名前。

`PROPERTY`

プロパティーの名前。

`PROTO`

プロパティーが属しているプロトコルの名前。現在サポートされているプロトコルは IPv4 と IPv6 です。

`PERM`

プロパティーの読み取り権と書き込み権。表示される値は、`r` (読み取り専用)、`w` (書き込み専用)、または `rw` (読み書き) です。

`CURRENT`

プロパティーの現在の値。値が設定されていないため無効になっているインタフェースの場合は、`--` と表示されます。

`PERSISTENT`

プロパティーの永続的な値。永続的な値は、リブート時に再度適用される値です。

`DEFAULT`

プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、`--` と表示されます。

`POSSIBLE`

プロパティーに設定できる値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形として *min* - *max* が表示される可能性があります。指定可能な値が不明な場合は `?` と表示され、指定可能な値が無限の場合は `--` と表示されます。

`-c`, `--parsable`

安定したマシン解析可能な形式を使用して表示します。このオプションには `-o` オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-p *prop*,..., --prop=*prop*
表示するプロパティーのコンマ区切りのリスト。サブコマンドの説明に続くインタフェースプロパティーの節を参照してください。

-m *protocol*, --module *protocol*
指定されたプロトコルに一致するプロパティーを表示します。有効な値は ipv4 と ipv6 です。

サポートされているインタフェースプロパティーの一覧については、後述の「インタフェースのプロパティー」を参照してください。

create-addr [-t] -T static [-d] -a {local | remote}=*addr*[/*prefixlen*],... *addrobj*
addrobj で指定されたインタフェースに IPv4 または IPv6 の静的アドレスを作成します。*addrobj* を作成するインタフェースはすでに存在している必要があります。作成された静的アドレスは *addrobj* で識別されます。

デフォルトでは、構成されたアドレスは「up」とマークされるため、送信パケットと受信パケットの発信元または着信先として使用できます。

すべてのアドレスオブジェクトが作成時に有効になります。アドレスオブジェクトの無効化および有効化の手順については、disable-addr サブコマンドと enable-addr サブコマンドを参照してください。

永続的な操作を一時オブジェクトに対して実行することはできません。つまり、一時的に作成されたインタフェースには、永続的なアドレスオブジェクトを作成することはできません。

addrobj 名に指定されたインタフェースが IPMP インタフェースであれば、IPMP インタフェース上に静的データアドレスが作成されます。*addrobj* 名に指定されたインタフェースが IPMP グループの基になるインタフェースであれば、基になるインタフェース上に静的テストアドレスが作成されます。

-t, --temporary
構成されたアドレスは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

-d, --down
構成されたアドレスを「down」とマークします。つまり、そのアドレスは IP パケットの発信元や着信先として使用されません。

-a {local | remote}=*addr*[/*prefixlen*],...
--address {local | remote}=*addr*[/*prefixlen*],...
addr は、ローカルエンドポイントまたはリモートエンドポイント (ポイントツーポイントインタフェースの場合) に対応する、リテラル IP アドレスまたはホスト名です。

ホスト名が指定されている場合は、その数値が /etc/hosts のエントリから一意に取得されます。このファイルに数値の IP アドレスが定義されていない場合は、nsswitch.conf(4) で hosts または ipnodes に指定されているリゾルバの

順序を使用して、数値が一意に取得されます。指定されたホスト名に複数のエントリが存在する場合は、エラーが生成されます。IP アドレスの作成はブートプロセスでネームサービスがオンラインになる前に行われるため、使用するホスト名がすべて `/etc/hosts` に含まれていることが重要です。

prefixlen がコマンド行で明示的に指定されていない場合、アドレスのネットマスクは次に示す順序で検索によって取得されます。

1. `nsswitch.conf(4)` で `netmasks` に指定されている順序を使用する
2. RFC 791 で定義されているクラスフルサブネット化セマンティクスを使用して IPv4 アドレスを解釈し、RFC 4291 の定義を使用して IPv6 アドレスを解釈します。

ポイントツーポイントインタフェースの場合は、ローカルエンドポイントのアドレスとともに、リモートエンドポイントのアドレスを指定する必要があります (例: `-a` local=*laddr*,remote=*raddr*)。リモートエンドポイントに *prefixlen* を指定すると、エラーが返されます。

インタフェースにローカルアドレスだけが必要な場合は、`-a` オプションに直接、`-a` *addr*[/*prefixlen*] のように指定します。アドレスは自動的にローカルアドレスとみなされます。

`create-addr` [`-t`] `-T` dhcp [`-w` *seconds* | forever] `-h` *hostname addrobj*

addrobj で指定されたインタフェースに、DHCP によって制御される IPv4 アドレスを作成します。作成された IPv4 アドレスは *addrobj* で識別されます。このコマンドの *addrobj* に、基になるインタフェースを指定するとテストアドレスが作成され、IPMP インタフェースを指定するとデータアドレスが作成されます。

すべてのアドレスオブジェクトが作成時に有効になります。アドレスオブジェクトの無効化および有効化の手順については、`disable-addr` サブコマンドと `enable-addr` サブコマンドを参照してください。

永続的な操作を一時オブジェクトに対して実行することはできません。つまり、一時的に作成されたインタフェースには、永続的なアドレスオブジェクトを作成することはできません。

addrobj 名に指定されたインタフェースが IPMP インタフェースであれば、DHCP によって取得されたアドレスが IPMP インタフェース上にデータアドレスとして作成されます。

`-h` *hostname*

クライアントが DHCP サーバーでクライアントのリース済み IPv4 アドレスをマップさせるホスト名を指定します。DHCP サーバーがホスト名の要求を満たせるという保証はありません。

`-t`, `--temporary`

構成されたアドレスは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

-w *seconds* | forever, --wait *seconds* | forever
処理が完了するまで待つ時間を秒単位で指定します。待機時間が指定されず、かつ処理内容がすぐに完了できない性質のものであった場合、ipadm はデフォルトで 120 秒間、要求された処理が完了するのを待ちます。デフォルトの待機時間は将来のリリースで変更される可能性があります。シンボリック値 forever も、リテラルな意味で使用できます。

create-addr [-t] -T addrconf [-i {local | remote}=*interface-id*] [-p {stateful | stateless}={yes | no},..] *addrobj*
addrobj で指定されたインタフェースに、自動設定 IPv6 アドレスを作成します。作成された IPv6 アドレスは *addrobj* で識別されます。

自動設定アドレスは、デフォルトのインタフェース ID を使用して生成されます (メディアタイプが Ethernet の場合、インタフェース ID はインタフェースの MAC アドレス)。この動作は -i オプションで上書きできます。

デフォルトでは、次のようになります。

- RFC 4862 に記述されているとおり、IPv6 アドレスは、ルーターで通知される接頭辞に基づいて自動設定されます。さらに、
- RFC 3315 に記述されているとおり、IPv6 アドレスは、DHCPv6 サーバーから提供される IPv6 アドレスを使用して、指定のインタフェース上に自動設定されます。(つまり、デフォルトのオプションは -p stateful=yes,stateless=yes です。)

すべてのアドレスオブジェクトが作成時に有効になります。アドレスオブジェクトの無効化および有効化の手順については、disable-addr サブコマンドと enable-addr サブコマンドを参照してください。

永続的な操作を一時オブジェクトに対して実行することはできません。つまり、一時的に作成されたインタフェースには、永続的なアドレスオブジェクトを作成することはできません。

addrobj 名に指定されたインタフェースが IPMP インタフェースであれば、IPv6 自動設定によって取得されたアドレスが IPMP インタフェース上にデータアドレスとして作成されます。

-t, --temporary
構成されたアドレスは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

-i {local | remote}=*interface-id*, --interface-id {local | remote}=*interface-id*
自動設定アドレスの生成に使用するインタフェース ID を指定します。

ポイントツーポイントインタフェースの場合は、リモートエンドポイントのインタフェース ID を指定できます (例: -i local=*lid*,remote=*rid*)。

インタフェースにローカルインタフェース ID だけが必要な場合は、-i オプションに直接、-i *lid* のように指定します。インタフェース ID は自動的にローカルインタフェース ID とみなされます。

-p {stateful | stateless}={yes | no},..
--prop {stateful | stateless}={yes | no},..
ステートフル自動設定方法とステートレス自動設定方法の一方または両方を有効にするかどうかを指定します。

-p stateful=no を指定すると、DHCPv6 で指定される IPv6 アドレスに基づいたステートフル自動設定は実行されません。

-p stateless=no を指定すると、ルーターで通知される接頭辞に基づいたステートレス自動設定は実行されません。

-p stateful=no,stateless=no を指定すると、どちらの方法の自動設定も実行されません。

-T addrconf オプションでは、デフォルトで -p stateful=yes,stateless=yes が使用されます。

delete-addr [-r] *addrobj*
addrobj で指定されたインタフェース上の、*addrobj* で識別されるアドレスをすべて削除します。これらのアドレスは永続的データストアからも削除されるため、リブート時にこれらのアドレスはインスタンス化されません。

アドレスオブジェクトが DHCP によって制御されるアドレスの場合、delete-addr は DHCP サーバーに通知することなくシステムからこのアドレスを削除し、あとで使用できるように現在のリースを記録します。

-r, --release
addrobj が DHCP によって制御されるアドレスの場合、このオプションを指定すると、インタフェース上の DHCP によって制御される IP アドレスがサーバーへの通知によって解放され、現在のリースが破棄されます。

show-addr [[-p] -o *field*[,...]] [-d] [*addrobj* | *interface/*]
指定された *addrobj* について、または指定されたインタフェース上に構成されているすべてのアドレスオブジェクト (永続的な構成だけに存在するアドレスオブジェクトも含む) について、アドレス情報を表示します。

-p, --parsable
安定したマシン解析可能な形式を使用して表示します。このオプションには -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-o *field*[,...], --output *field*[,...]
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。*field* 名には、次に示すフィールドのいずれか、あるいは、すべての

フィールドを表示するための特殊な値 all を指定する必要があります。インタフェースごとに、次のフィールドを表示できます。

ADDROBJ
: アドレスオブジェクトの名前。

TYPE
: アドレスオブジェクトのタイプ。from-gz、static、dhcp、または addrconf のいずれかです。static、dhcp、および addrconf タイプは、create-addr の -T オプションで指定されたアドレスオブジェクトのタイプに対応します。from-gz タイプは、非大域ゾーンでのみ表示され、非大域排他的 IP ゾーンに設定されている allowed-address プロパティーに基づいて大域ゾーンから構成されたアドレスであることを示します。

STATE
: アドレスオブジェクトの状態。このフィールドは、-o に all を指定した場合だけ表示されます。次のいずれかの値が表示されます。

disabled
: アドレスはアクティブな構成に含まれています (disable-addr と disable-if を参照)。

down
: アドレスは管理上停止されています (down-addr を参照)。

duplicate
: アドレスは、別のシステムの IP アドレスと競合していることが重複アドレス検出 (DAD) で見つかったため、競合が解決されるまで使用できません。システムでは、競合が解決されたかどうかを調べるために、定期的に DAD が実行されます。または、refresh-addr を使用して DAD をただちに再実行することもできます。

inaccessible
: アドレスが構成されている IP インタフェースに障害が発生したため、アドレスは使用できません。

ok
: アドレスは有効で、稼働しており、正しく動作しています。システムは、このアドレス宛の IP パケットを受け入れます。また、設定されている IP 発信元アドレス選択ポリシーに従い、このアドレスを使用して IP パケットを発信します。

tentative
: 現在、アドレスには重複アドレス検出が行われています (たとえば、up-addr または refresh-addr の一環として)。

CURRENT
: アクティブな構成のアドレスオブジェクトに関して、次のフラグのいずれかを示します。このフィールドはデフォルトでは表示されず、-o で all または current を指定した場合のみ表示されます。

d (deprecated)
このアドレスが送信パケットの発信元アドレスとして使用されるのは、そのインタフェース上で利用可能なアドレスがほかに存在しない場合と、アプリケーションがこのアドレスに明示的にバインドした場合だけです。

p (private)
このアドレスはルーティングデーモンによって通知されません。

t (temporary)
RFC 3041 で定義されている一時 IPv6 アドレスです。

U (up)
アドレスは「up」とマークされ、送信/受信パケットの発信元/着信先として使用できます。

u (unnumbered)
アドレスは、システム内のほかのリンクのローカルアドレスに一致しています。

PERSISTENT
アドレスオブジェクトがリブート時にインスタンス化されるとき、または enable-addr サブコマンドで再度有効化されるときに適用する構成を指定します。有効な値は、U、p、および d の任意の組み合わせです (前述の説明を参照)。

ADDR
数値の IPv4 または IPv6 アドレス。ポイントツーポイントインタフェースの場合は、両方のエンドポイントのアドレスが表示されます (*laddr*-->*raddr*)。dhcp タイプのアドレスオブジェクトの場合は、アドレスオブジェクトの状態が disabled であるか、アドレスが IPv4 アドレス 0.0.0.0 または IPv6 アドレス :: であれば、疑問符 (?) が表示されます。

CID-TYPE
アドレスが DHCP を使用して取得されている場合は、dhcpagent(1M) で使用されるクライアント ID のタイプ。IPv4 の場合は、RFC 4361 のクライアント ID の構築に使用される DUID のタイプが表示されます。タイプは DUID-LLT、DUID-EN、DUID-LL、other、または default のいずれかです。このフィールドはデフォルトの出力には表示されません。表示するには、-d を使用するか、-o に cid-type または all を使用します。

DUID-LLT
タイプ 1 の RFC 3315 DUID が CID-VALUE の構築に使用されます (例: 1,1,63463777,0a:0b:0c:0d:0e:0f)。詳細については、RFC を参照してください。

DUID-EN
タイプ 2 の RFC 3315 DUID が CID-VALUE の構築に使用されます (例: 1,1,63463777,0a:0b:0c:0d:0e:0f)。詳細については、RFC を参照してください。

DUID-LL
タイプ 3 の RFC 3315 DUID が CID-VALUE の構築に使用されます (例: 1,1,63463777,0a:0b:0c:0d:0e:0f)。詳細については、RFC を参照してください。

other
タイプ {0,4-65535} の RFC 3315 DUID がクライアント ID の生成に使用されます (例: 4,0x734633)。または、CID-VALUE は、RFC 3315 に準拠していないクライアント ID そのものです (例: Sun、0xab3146)。

default
クライアント ID の構築に RFC 3315 DUID は使用されていないことを示します。代わりに、RFC 2132 に従い、インタフェースの MAC アドレスを使用してクライアント ID が生成されます。CID-VALUE は、文字列 0x01 とそれに続く MAC アドレスの 16 進文字列で構成されます。これは IPv4 の場合にのみ当てはまります。

CID-VALUE
アドレスが DHCP を使用して取得されている場合は、dhcpagent(1M) で使用されるクライアント ID の値。使用される形式は、/etc/default/dhcpagent ファイル内の構成パラメータ CLIENT_ID の形式に従います。dhcpagent(1M) の CLIENT_ID の説明を参照してください。CID-TYPE が default の場合、CID-VALUE は、RFC 2132 に従って構築されたレガシー CLIENT-ID になります。このフィールドはデフォルトの出力には表示されません。表示するには、-d を使用するか、-o に cid-type または all を使用します。

BEGIN
DHCP を使用して取得されたアドレスの場合、リースが開始された時刻 (存在する場合)。時刻は、LC_TIME ロケール環境変数で規定される形式で表示されます。DHCP によって構成されないアドレス、あるいは、まだリースが取得されていない DHCP アドレスの場合は、-- (2 つのハイフン) が表示されます。このフィールドはデフォルトの出力には表示されません。表示するには、-d を使用するか、-o に cid-type または all を使用します。

EXPIRE
DHCP を使用して取得されたアドレスの場合、リースが期限切れになる時刻 (存在する場合)。時刻は、LC_TIME ロケール環境変数で規定される形式で表示されます。DHCP によって構成されないアドレス、あるいは、まだリースが取得されていない DHCP アドレスの場合は、-- (2 つのハイフン) が表示されます。このフィールドはデフォルトの出力には表示されません。表示するには、-d を使用するか、-o に cid-type または all を使用します。

RENEW
DHCP を使用して取得されたアドレスの場合、リースが最後に更新された時刻。時刻は、LC_TIME ロケール環境変数で規定される形式で表示されます。DHCP によって構成されないアドレス、あるいは、まだリースが取得

されていない DHCP アドレスの場合は、-- (2 つのハイフン) が表示されます。このフィールドはデフォルトの出力には表示されません。表示するには、-d を使用するか、-o に cid-type または all を使用します。

-d, --dhcp
DHCP を使用して取得されたアドレスの dhcp ステータスフィールドを表示します。表示されるフィールドは、ADDROBJ、STATE、ADDR、CID-TYPE、CID-VALUE、BEGIN、EXPIRE 、および RENEW です。このオプションの出力は人間が読める形式だけです。このオプションを -p と組み合わせて使用することはできません。

注 – e1000g1/_a や e1000g1/_b のように、アドレスオブジェクトの名前が下線 (_) で始まる場合もあります。このような名前は、ipadm プログラミングライブラリを使用するアプリケーションのために自動的に生成されたものです。アドレスのアドレスオブジェクト名に疑問符 (?) が含まれている場合もあります。これは、そのアドレスが ipadm ライブラリの外部で作成されたものであり、ipadm には認識されていないことを意味します。

down-addr [-t] *addrobj*
addrobj で識別されるアドレスを「down」とマークして、送信/受信パケットの発信元/着信先として使用できないようにします。このコマンドは、down-addr の呼び出しの前にすでに「down」とマークされていたアドレスオブジェクトには影響しません。アドレスオブジェクトが addrconf タイプの場合、このコマンドはエラーを返します。

-t, --temporary
構成されたアドレスは一時的なもので、変更はアクティブな構成だけに適用されることを示します。アドレスオブジェクトのタイプが dhcp の場合、このオプションは必須です。

up-addr [-t] *addrobj*
addrobj で識別されるアドレスを「up」とマークして、送信/受信パケットの発信元/着信先として使用できるようにします。このサブコマンドは、重複アドレスなのでシステムによって「down」とマークされているアドレスオブジェクトや、up-addr の呼び出しの前にすでに「up」とマークされていたアドレスオブジェクトには影響しません。アドレスオブジェクトが addrconf タイプの場合、このコマンドはエラーを返します。

-t, --temporary
構成されたアドレスは一時的なもので、変更はアクティブな構成だけに適用されることを示します。アドレスオブジェクトのタイプが dhcp の場合、このオプションは必須です。

refresh-addr [-i] *addrobj*
addrobj が static タイプの場合、このアドレスオブジェクトで識別されるアドレスに対し、必要に応じて DAD (重複アドレス検出) が再開されます。

addrobj が dhcp タイプの場合、このアドレスに取得されたリース期間が DHCP クライアントデーモンによって延長されます。

addrobj が addrconf タイプの場合、このコマンドはエラーを返します。

-i, --inform
指定された IP アドレスについて、DHCP からネットワーク構成パラメータを取得します。その際、アドレスのリースは取得しません。これは、DHCP 以外のメカニズムを使って IP アドレスを取得する場合に役立ちます。

disable-addr -t *addrobj*
アドレスをアクティブな構成から削除することによって無効にします。もともと永続的なアドレスオブジェクトとして作成されていた場合、永続的な構成は変更されません。この *addrobj* を再度有効にするには、enable-addr を使用するようにしてください。

-t, --temporary
この無効化は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

enable-addr -t *addrobj*
構成を永続的ストアから読み取ることで、指定された *addrobj* を有効にします。アドレスオブジェクトには、永続的なアドレスプロパティーがすべて適用されます。このサブコマンドでは、アドレスオブジェクトを有効にするインタフェースが存在する必要があります。インタフェース自体がアクティブな構成には存在せず永続的ストアには存在する場合、つまりインタフェースが無効になっている場合は、enable-addr を呼び出す前に enable-if を実行する必要があります。

-t, --temporary
この有効化は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

set-addrprop [-t] -p prop=*value*[,...] *addrobj*
指定された *addrobj* にプロパティーの値を設定します。*addrobj* が複数のアドレスに対応している場合は、*addrobj* で参照されるすべてのアドレスにプロパティーの変更が適用されます。一度に指定できるプロパティーは 1 つだけです。*addrobj* でサポートされているプロパティーおよびその指定可能な値は、show-addrprop サブコマンドで取得できます。*addrobj* が addrconf タイプの場合、このコマンドはエラーを返します。

-t, --temporary
この変更は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

-p prop=*value*[,...], -–prop prop=*value*[,...]
指定の値に設定するプロパティー。

`reset-addrprop` [`-t`] `-p` *prop addrobj*

指定されたアドレスのプロパティーをデフォルト値にリセットします。`-t` を省略すると、プロパティーの永続的な値はすべて削除されます。一度に変更できるプロパティーは 1 つだけです。*addrobj* が `addrconf` タイプの場合、このコマンドはエラーを返します。

`-t`, `--temporary`

このリセットは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`-p` *prop*, `--prop` *prop*

リセットするプロパティー。

`show-addrprop` [[`-c`] `-o` *field*[,...]] [`-p` *prop*,...] [*addrobj*]

構成されているすべてのアドレスオブジェクトまたは特定の *addrobj* について、1 つまたは複数のプロパティーの現在の値と永続的な値を表示します。一度に複数のプロパティーを取得するには、プロパティー名をコンマで区切って `-p` オプションに指定します。`-p` オプションを指定しない場合、 使用可能なすべてのプロパティーが表示されます。*addrobj* が `addrconf` タイプの場合、このコマンドはエラーを返します。

`-o` *field*[,...], `--output` *field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。*field* 名には、次に示すフィールドのいずれか、あるいは、すべてのフィールドを表示するための特殊な値 `all` を指定する必要があります。*addrobj* ごとに、次のフィールドを表示できます。

`ADDROBJ`

アドレスオブジェクトの名前。

`PROPERTY`

プロパティーの名前。

`PERM`

プロパティーの読み取り権と書き込み権。表示される値は、`r` (読み取り専用)、`w` (書き込み専用)、または `rw` (読み取り/書き込み) です。

`CURRENT`

プロパティーの現在の値。値が設定されていないため無効になっているアドレスの場合は、2 つのハイフン (`--`) が表示されます。

`PERSISTENT`

プロパティーの永続的な値。永続的な値は、リブート時に再度適用される値です。

`DEFAULT`

プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、2 つのハイフン (`--`) が表示されます。

POSSIBLE
プロパティーに設定できる値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形として *min* - *max* が表示される可能性があります。指定可能な値が不明な場合は疑問符 (?) が表示され、指定可能な値が無限の場合は 2 つのハイフン (--) が表示されます。

-c, --parsable
安定したマシン解析可能な形式を使用して表示します。このオプションには -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-p *prop*,..., --prop=*prop*
表示するプロパティーのコンマ区切りのリスト。サブコマンドの説明に続くアドレスオブジェクトのプロパティーに関する節を参照してください。

set-prop [-t] -p prop[+ | –]=*value*[,...] *protocol*
プロトコルのプロパティーの値を、指定された *value* に変更します。プロパティーが複数の値を取る場合は、区切り文字としてコンマを使用して値を指定します。一度に指定できるプロパティーは 1 つだけです。デフォルトでは、値は永続的で、リブート時に再度適用されます。プロトコルでサポートされているプロパティーおよびその指定可能な値は、show-prop サブコマンドで取得できます。

サポートされているプロトコルは、ip、ipv4、ipv6、icmp、tcp、udp、および sctp です。

一部のプロパティーでは、プロパティーの値をグローバルに設定することもインタフェースごとに設定することもできます。インタフェースごとの値は set-ifprop サブコマンドで設定できます。そのような場合、管理者がプロパティーのインタフェースごとの値をカスタマイズして、グローバルな値とは異なる値にすると、そのインタフェースではグローバルな設定よりもインタフェースごとの値が優先されます。

-t, --temporary
このプロパティーの変更は一時的なもので、変更はアクティブな構成だけに適用されることを示します。

-p prop[+|-]=*value*[,...], --prop prop[+|-]=*value*[,...]
指定の値に設定するプロパティー。次の「修飾子」も提供されているため、割り当て操作に加え、追加操作と削除操作も実行できます。

+
指定された値を現在の値リストに追加します。

-
指定された値を現在の値リストから削除します。

=
新しい割り当てを行い、現在の値はすべて削除します。

修飾子の使用方法の詳細については、「使用例」を参照してください。

`reset-prop` [`-t`] `-p` *prop protocol*

指定されたプロトコルのプロパティーを、そのプロパティーのデフォルト値にリセットします。-t を使用しない場合、プロパティーの永続的な値はすべて削除されます。一度に変更できるプロパティーは 1 つだけです。

`-t`, `--temporary`

このリセットは一時的なもので、変更はアクティブな構成だけに適用されることを示します。

`-p` *prop*, `-–prop` *prop*

リセットするプロパティー。

`show-prop` [[`-c`] `-o` *field*[,...]] [`-p` *prop*[,...] *protocol* | *protocol*]

サポートされているすべてのプロトコルまたは特定のプロトコルについて、1 つまたは複数のプロパティーの現在の値と永続的な値を表示します。一度に複数のプロパティーを取得するには、プロパティー名をコンマで区切って `-p` オプションに指定します。`-p` オプションを指定しない場合、 使用可能なすべてのプロパティーが表示されます。

`-o` *field*[,...], `--output` *field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。*field* 名には、次に示すフィールドのいずれか、あるいは、すべてのフィールドを表示するための特殊な値 `all` を指定する必要があります。各 *protocol* について、次のフィールドを表示できます。

`PROTO`

プロトコルの名前。

`PROPERTY`

プロパティーの名前。

`PERM`

プロパティーの読み取り権と書き込み権。表示される値は、`r` (読み取り専用)、`w` (書き込み専用)、または `rw` (読み取り/書き込み) です。

`CURRENT`

プロパティーの現在の値。値が設定されていないため無効になっているアドレスの場合は、2 つのハイフン (`--`) が表示されます。値が不明な場合は疑問符 (`?`) が表示されます。プロパティーの現在の値が `POSSIBLE` 値のリストに含まれていない場合は、キーワード `custom` が表示されます。

`PERSISTENT`

プロパティーの永続的な値。永続的な値は、リブート時に再度適用される値です。

`DEFAULT`

プロパティーのデフォルト値。プロパティーにデフォルト値がない場合は、2 つのハイフン (`--`) が表示されます。

POSSIBLE
set-prop サブコマンドで使用するプロパティー設定の値のコンマ区切りのリスト。値に数値の範囲がある場合は、短縮形として *min* - *max* が表示される可能性があります。指定可能な値が不明な場合は疑問符 (?) が表示され、指定可能な値が無限の場合は 2 つのハイフン (--) が表示されます。

-c, --parsable
安定したマシン解析可能な形式を使用して表示します。このオプションには -o オプションが必要です。下記の「解析可能な出力形式」を参照してください。

-p *prop*,..., --prop=*prop*
表示するプロパティーのコンマ区切りのリスト。サブコマンドの説明に続くプロトコルのプロパティーに関する節を参照してください。

各プロトコルでサポートされているプロパティーの一覧については、後述の「プロトコルのプロパティー」を参照してください。

解析可能な出力形式

ipadm の一連の「show」サブコマンドには、マシン解析可能な形式で出力を表示する -o オプションがあります。この出力形式は、コロン (:) で区切られたフィールドで構成された 1 つ以上の行です。表示されるフィールドは、使用されているサブコマンドに固有のもので、そのサブコマンドの -o オプションに基づいて一覧表示されます。出力には、-o オプションを使用して要求したフィールドのみが、要求した順序で表示されます。サブコマンドのすべてのフィールドを表示する -o all オプションは、解析可能な出力には使用できません。

複数のフィールドを要求した場合、すべてのコロン文字は、バックスラッシュ (\) でエスケープされてから出力されます。同様に、バックスラッシュ文字もエスケープされます (\\)。このエスケープ形式は、環境変数 IFS=: を設定したシェルの read(1) 関数を使って解析できます。ただし、1 つのフィールドだけを要求した場合はエスケープは行われません。

プロトコルのプロパティー

次のプロトコルプロパティーがサポートされています。

注 – プロトコルに固有の、「_」(下線) で始まるプロトコルプロパティーがあります。これらのプロパティーはデフォルトで変更または削除されることがあり、ipadm show-prop 出力には表示されません。詳細は、『Oracle Solaris Tunable Parameters Reference Manual』を参照してください。

cong_default (TCP, SCTP)
新しい接続の作成時にプロトコルによって使用されるデフォルトの輻輳制御アルゴリズムを指定します。あとで接続の有効期間内にアプリケーションで別のアルゴリズムを選択することもできます。デフォルトとして設定できるのは有効になっているアルゴリズムのみです (cong_enabled を参照)。

cong_enabled (TCP, SCTP)
このオプションを使用すると、輻輳制御アルゴリズムを有効または無効にすることができます。デフォルトでは、システムにインストールされているすべてのア

ルゴリズムが有効になります。無効になっているアルゴリズムをデフォルトとして設定したり (cong_default を参照)、アプリケーションで使用したりすることはできません。

アルゴリズムの追加または削除には、set-prop サブコマンドと修飾子 + および - を使用します。

ecn (TCP)
明示的な輻輳制御 (詳細は、RFC 3168 を参照)。指定可能な値は、上記と同じ never、passive、および active です。

extra_priv_ports (TCP, SCTP, UDP)
このオプションは、1-1023 の範囲外にある追加の特権ポートを定義します。このリストに含まれるポートにバインドしようとするプログラムは、root ユーザーとして実行する必要があります。これにより、通常ユーザーが特定のポートでサーバープロセスを開始することを防止します。

これらのポートの追加、削除、および割り当てを行うには、set-prop サブコマンドと修飾子 +、-、および = を使用します。使用方法については、後述の「使用例」を参照してください。

forwarding (IPv4), forwarding (IPv6)
グローバルな IPv4 転送または IPv6 転送を有効または無効にします。構成されているすべてのインタフェースがパケットの転送を開始または停止します。set-ifprop による個々のインタフェースの設定は、このグローバルなオプションよりも優先されます。

hostmodel (IPv4), hostmodel (IPv6)
マルチホームシステムでの IP パケットの送受信動作を制御します。hostmodel の値は strong または weak に設定できます。これは、RFC 1122 の同等のエンドシステムモデル定義に対応します。さらに、3 つ目の値 src-priority もサポートされています。src-priority hostmodel シナリオでは、パケットの着信先 IP アドレスがホストのいずれかのインタフェース上に構成され「UP」とマークされていれば、パケットは任意のインタフェースで受け入れられます。パケットの送信時、パケットの IP 着信先への経路が複数ある場合は、パケットの IP 発信元アドレスが送信インタフェース上に構成されている経路が優先されます。そのような経路がない場合、弱い ES の場合と同様に、フォールバックによって「最適な」経路が選択されます。

max_buf (TCP, SCTP, UDP, ICMP)
ソケットの送信バッファーまたは受信バッファーの最大サイズ。このプロパティーの現在の値によって、recv_buf と send_buf の最大値が制限されます。

recv_buf (TCP, SCTP, UDP, ICMP)
send_buf (TCP, SCTP, UDP, ICMP)
指定されたプロトコルの受信バッファーサイズまたは送信バッファーサイズを変更します。これらのプロパティーの最大値は、max_buf プロパティーの現在の値に制約されます。

`sack` (TCP)
選択的肯定応答 (SACK) を使用すると、受信者は順序の誤ったデータを選択的に受け入れることができます。損失の多いリンクでデータ転送のパフォーマンスを向上させるためのものです。SACK については、RFC 2018 を参照してください。指定可能な値とその意味は次のとおりです。

`never`
SACK を受け入れず、SACK 情報の送信も行いません。

`passive`
SACK を受け入れますが、送信はしません。

`active`
SACK を受け入れ、SACK 情報の送信も行います。

`smallest_anon_port` (TCP, SCTP, UDP)
`largest_anon_port` (TCP, SCTP, UDP)
これらのオプションは、エフェメラルポートの上限と下限を定義します。エフェメラル (「短命な」の意) ポートは、発信ネットワーク接続を確立するときに使用されます。`smallest_anon_port` の現在の値は、常に `largest_anon_port` の現在の値以下になるようにしてください。

`smallest_nonpriv_port` (TCP, SCTP, UDP)
このオプションは、非特権ポートの開始を定義します。通常、非特権ポートの範囲は 1024 から始まります。非特権ポートにバインドしようとするプログラムは、root ユーザーとして実行する必要はありません。

`ttl` (IPv4), `hoplimit` (IPv6)
IPv4 ヘッダーまたは IPv6 ヘッダーの `ttl/hoplimit` フィールドに設定する値を指定します。システムが *N* ホップより遠くにあるほかのシステムに到達することを防止するために使用できます。この *N* は、指定した値です。

インタフェースのプロパティー

次のインタフェースプロパティーがサポートされています。

`arp`
インタフェースでのアドレス解決プロトコル (ARP) の使用を有効または無効にします。ARP は、ネットワークレベルのアドレスとリンクレベルのアドレスとの間のマッピングに使用されます。これは現時点では、IPv4 アドレスと MAC アドレス間のマッピング用として実装されています。指定可能な値は、`on` または `off` です。デフォルトは `on` です。

`exchange_routes`
このインタフェースでの経路指定情報の交換を有効または無効にします。指定可能な値は、`on` または `off` です。デフォルトは `off` です。

`group`
このインタフェースが基になるインタフェースとして属している IPMP インタフェースのグループ名を指定します。インタフェースのクラスが IPMP の場

合、これはIPMPグループの名前を指定します。このプロパティーはIPMPインタフェースでのみ読み書き可能です。ほかのインタフェースクラスの場合、このプロパティーは読み取り専用です。

`forwarding`
インタフェースでのIP転送を有効または無効にします。有効になっている場合、このインタフェースへの、およびこのインタフェースからの、IPパケットの転送を行えます。指定可能な値は、`on`または`off`です。デフォルトは`off`です。

`metric`
インタフェースの経路指定メトリックを*n*に設定します。値が指定されなかった場合のデフォルトは、`0`になります。経路指定メトリックは、経路指定プロトコルによって使用されます。メトリックが高いほど、その経路は好まれません。メトリックは、着信先となるネットワークまたはホストへの加算ホップとしてカウントされます。

`mtu`
インタフェースの最大転送単位を*n*に設定します。多くのネットワークタイプでは、MTUには上限があります。たとえば、Ethernetの場合は1500です。

`nud`
特定のポイントツーポイント物理インタフェース上で近傍不到達検出メカニズムを有効または無効にします。指定可能な値は、`on`または`off`です。デフォルトは`on`です。

`standby`
このインタフェースをIPMPグループの予備インタフェースとして構成するかどうかを指定します。このプロパティーはIPMPインタフェースには適用できません。

`usesrc`
特定の物理インタフェースまたは仮想インタフェースを発信元アドレス選択用として指定します。キーワード`none`を使用した場合、それまでの選択がすべてクリアされます。デフォルトは`none`です。

アドレスのプロパティー

次のアドレスプロパティーがサポートされています。`addrconf`アドレスオブジェクトのアドレスプロパティーの変更はサポートされていません。

`deprecated`
新しい通信ではこのアドレスを発信元アドレスとして使用しないようにしてください。ただし、このアドレス宛のパケットは期待どおりに処理されます。指定可能な値は、`on`または`off`です。デフォルトは`off`です。このプロパティーは`dhcp`タイプのアドレスオブジェクトではサポートされていません。

`prefixlen`
アドレスの左から何ビットがアドレスのIPv6接頭辞またはIPv4ネットマスクに該当するかを指定します。残りの下位ビットは、アドレスのホスト部を定義します。`prefixlen`をアドレスのテキスト表現に変換すると、アドレスのネット

ワーク部として使用するビット位置には 1 が、ホスト部として使用するビット位置には 0 が含まれています。prefixlen は単一の 10 進数で指定する必要があります。このプロパティーは dhcp タイプのアドレスオブジェクトではサポートされていません。

private
アドレスを in.routed ルーティングデーモンで通知しないように指定します。指定可能な値は、on または off です。デフォルトは off です。

reqhost
クライアントが DHCP サーバーでクライアントのリース済み IPv4 をマップさせるホスト名。ホスト名の要求が満たされる保証はありません。

transmit
アドレスオブジェクトで参照されるアドレスを使用してパケットを送信できるようにします。これは、アドレスが「up」状態にある場合のデフォルト動作です。指定可能な値は、on または off です。デフォルトは on です。

zone
アドレスオブジェクトで参照されるすべてのアドレスを配置するゾーンを指定します。指定されたゾーンは、カーネル内でアクティブになっており、準備完了か実行中の状態になっていなければいけません。ゾーンが停止またはリブートすると、インタフェースは unplumb されます。ゾーンは共有 IP ゾーンとして構成されている必要があります。排他的 IP ゾーンにネットワークインタフェース名を割り当てるには、zonecfg(1M) を使用します。ゾーンの割り当てがリブート後も持続するように変更するには、zonecfg(1M) を使用してください。指定可能な値は、システム上に構成されているすべてのゾーンのリストです。デフォルトは global です。

オペランド

各 ipadm サブコマンドは、次のいずれかのオブジェクトに対して動作します。

addrobj
ネットワークインタフェース上に構成されているアドレスは *addrobj* で識別されます。*addrobj* は 2 つの部分から成ります。最初の部分は、アドレスが構成されているネットワークインタフェースの名前です。2 つ目の部分は、ユーザーが指定した文字列です。任意の英数字を 32 文字まで使用でき、英文字で始まる必要があります。*addrobj* の 2 つの部分はスラッシュ (/) で区切られます。アドレスオブジェクトは常に、システム内にあるアドレスの一意のセットを表します。

interface
ネットワークアドレスが構成されているネットワークインタフェースの名前。一般に、この名前には任意の英数字に加えて下線 (_) とピリオド (.) も使用できますが、英字で始まり、数字で終わる必要があります。

protocol
プロパティーを設定する TCP/IP インターネットプロトコルファミリの名前。サポートされているプロトコルは、ip、ipv4、ipv6、icmp、tcp、sctp、および udp です。

使用例

例 1　IPv4 静的アドレスの作成

次のコマンドは、インタフェース bge1 (リンク名 net1) 上にアドレス 10.2.3.4/24 を作成し、そのアドレスを使用するために「up」とマークします。

```
# ipadm create-ip net1
# ipadm create-addr -T static -a local=10.2.3.4/24 net1/v4static1
```

次のコマンドは、インタフェース net1 上に別のアドレス 10.2.3.5/24 を作成しますが、明示的に「up」とマークされるまでアドレスを「down」とマークします。

```
# ipadm create-addr -T static -d -a 10.2.3.5/24 net1/v4static2
```

local が使用されておらず、指定されたアドレスが 1 つだけなので、10.2.3.5/24 はローカルアドレスとみなされます。

次のコマンドは、以前に「down」とマークされたアドレスオブジェクト net1/v4static2 を「up」とマークします。

```
# ipadm up-addr net1/v4static2
```

アドレスオブジェクトに DUPLICATE フラグが設定されていた場合は、アドレスがネットワーク上でまだ重複しているかどうかを refresh-addr で確認します。そうでない場合、アドレスは「up」とマークされます。

```
# ipadm refresh-addr net1/v4static2
```

次のコマンドは、構成されているアドレスを一覧表示します。これは、アドレス net1/v4static2 が重複していないことを示します。

```
# ipadm show-addr
ADDROBJ           TYPE    STATE      ADDR
lo0/v4            static  ok         127.0.0.1/8
lo0/v6            static  ok         ::/128
net1/v4static1    static  ok         10.2.3.4/24
net1/v4static2    static  ok         10.2.3.10/24
```

例 2　DHCPv4 によって制御されるアドレスの作成

次のコマンドは、インタフェース bge1 (リンク名 net1) 上の DHCPv4 アドレスを取得します。

```
# ipadm create-ip net1
# ipadm create-addr -T dhcp net1/dhaddr
# ipadm show-addr net1/dhaddr
ADDROBJ           TYPE    STATE      ADDR
net1/dhaddr       dhcp    ok         10.8.48.173/25
```

次のコマンドは、DHCPv4 アドレスオブジェクト net1/dhaddr のリース期間を延長します。

```
# ipadm refresh-addr net1/dhaddr
```

例 3　IPv6 アドレスの作成

次の一連のコマンドは、in.ndpd をデフォルトのインタフェース ID とともに使用して、IPv6 アドレスを bge1 (リンク名 net1) 上に自動設定します。リンクローカルアドレスが最初に設定されたあと、ステートレスおよびステートフルな自動設定アドレスが in.ndpd によって追加されます。

```
# ipadm create-ip net1
# ipadm create-addr -T addrconf net1/v6addr
```

次のコマンドは、IPv6 静的アドレスを作成します。リンクローカルアドレスでない IPv6 アドレスを構成するには、インタフェース上にリンクローカルアドレスがすでに構成されている必要があります。これは、前の手順の -T addrconf で達成されています。

```
# ipadm create-addr -T static -a local=2ff0::f3ad/64 net1/v6static
```

次のコマンドは、IPv6 アドレスの接頭辞の長さを変更します。

```
# ipadm set-addrprop -p prefixlen=80 net1/v6static
```

すべての自動設定アドレスおよび更新された接頭辞の長さは、アドレスを一覧表示することで確認できます。

```
# ipadm show-addr
ADDROBJ       TYPE     STATE     ADDR
lo0/v4        static   ok        127.0.0.1/8
lo0/v6        static   ok        ::/128
net1/v6addr   addrconf ok        fe80::203:baff:fe94:2f01/10
net1/v6addr   addrconf ok        2002:a08:39f0:1:203:baff:\
                                            fe94:2f00/64
net1/v6addr   addrconf ok        2001:db8:1:2::402f/128
net1/v6static static   ok        2ff0::f3ad/80
```

例 4　IPv4 トンネルの構成

次に示す最初のコマンド (ipadm) は、トンネルの発信元アドレスを作成します。次に、dladm コマンドでトンネルリンクが作成されます。最後の ipadm コマンドは、トンネル IP インタフェース上に IPv4 アドレスと IPv6 アドレスを構成します。

```
# ipadm create-ip net1
# ipadm create-addr -T static -a local=10.2.3.4/24 net1/v4static
# dladm create-iptun -T ipv4 -a local=10.2.3.4,remote=10.2.3.5 tun0
# ipadm create-ip tun0
# ipadm create-addr -T static \
       -a local=173.129.134.1,remote=173.129.134.2 tun0/v4tunaddr
# ipadm create-addr -T static \
       -a local=2ff1::3344,remote=2ff1::3345 tun0/v6tunaddr
# ipadm show-addr
ADDROBJ        TYPE   STATE     ADDR
lo0/v4         static ok        127.0.0.1/8
```

例 4 IPv4 トンネルの構成 (続き)

```
lo0/v6         static ok        ::/128
net1/v4static  static ok        10.2.3.4/24
tun0/v4tunaddr static ok        173.129.134.1-->173.129.134.2
tun0/v6tunaddr static ok        2ff1::3344-->2ff1::3345
```

例 5 すべてのインタフェースの表示

次のコマンドを使用すると、すべてのインタフェースを表示できます。

```
# ipadm show-if -o all
IFNAME  CLASS    STATE    ACTIVE CURRENT       PERSISTENT OVER
lo0     loopback ok       yes    -m-v------46  --46       --
net0    ip       ok       yes    bm--------46  --46       --
e1000g0 ip       ok       yes    bm---l----46  -l46       --
e1000g1 ip       ok       yes    bm---l----46  -l46       --
ipmp0   ipmp     down     yes    bm--------46  --46       e1000g0 e1000g1
tun0    ip       failed   no     -mp-------46  --46       --
vni0    vni      disabled no     bm-v--------  --46       --
```

例 6 インタフェースのプロパティーの表示

次のコマンドは、指定されたインタフェースのすべてのインタフェースプロパティーを表示します。

```
# ipadm show-ifprop net0
IFNAME PROPERTY        PROTO PERM CURRENT PERSISTENT DEFAULT  POSSIBLE
net0   arp             ipv4  rw   on      --         on       on,off
net0   forwarding      ipv4  rw   off     on         off      on,off
net0   metric          ipv4  rw   2       2          0        --
net0   mtu             ipv4  rw   1500    --         1500     68-1500
net0   exchange_routes ipv4  rw   off     --         off      on,off
net0   usesrc          ipv4  rw   none    --         none     --
net0   forwarding      ipv6  rw   off     --         off      on,off
net0   metric          ipv6  rw   2       2          0        --
net0   mtu             ipv6  rw   1500    --         1500     1280-1500
net0   nud             ipv6  rw   on      --         on       on,off
net0   exchange_routes ipv6  rw   off     on         off      on,off
net0   usesrc          ipv6  rw   none    --         none     --
net0   group           ip    rw   grp0    --         --       --
net0   standby         ip    r-   off     --         off      on,off
```

例 7 インタフェースごとのプロパティーの構成

次のコマンドは、インタフェース net0 の IPv4 MTU を 900 に設定します。

```
# ipadm set-ifprop -m ipv4 -p mtu=900 net0
```

次のコマンドは、インタフェース net0 の IPv6 MTU を 1400 に設定します。

```
# ipadm set-ifprop -m ipv6 -p mtu 1400 net0
```

例 7 インタフェースごとのプロパティーの構成 (続き)

結果を確認します。

```
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY        PROTO PERM CURRENT PERSISTENT DEFAULT  POSSIBLE
net0   mtu             ipv4  rw   900     900        1500     68-1500
net0   mtu             ipv6  rw   1400    1400       1500     1280-1500

# ipadm show-ifprop -m ipv6 -p mtu net0
IFNAME PROPERTY        PROTO PERM CURRENT PERSISTENT DEFAULT  POSSIBLE
net0   mtu             ipv6  rw   1400    1400       1500     1280-1500
```

例 8 サポートされているプロパティーの表示

次のコマンドは、TCP でサポートされているプロパティーを表示します。

```
# ipadm show-prop tcp
PROTO PROPERTY              PERM CURRENT PERSISTENT DEFAULT   POSSIBLE
tcp   ecn                   rw   active  active     passive   never,passive,
                                                               active
tcp   extra_priv_ports      rw   --      1,65535    2049,4045 1-65535
tcp   largest_anon_port     rw   32768   32768      65535     1024-65535
tcp   sack                  rw   active  --         active    never,passive,
                                                               active
tcp   recv_buf              rw   29567   --         49152     2048-1073741824
tcp   send_buf              rw   21354   --         49152     4096-1073741824
tcp   max_buf               ro   65536   --         32768     4096-1073741824
tcp   smallest_anon_port    rw   32768   --         32768     1024-65535
tcp   smallest_nonpriv_port rw   1024    --         1024      1024-32768
```

例 9 グローバルな IPv4 転送の構成

次の一連のコマンドは、グローバルな IPv4 転送を構成し、その設定をインタフェース net0 については上書きします。

```
# ipadm set-prop -p forwarding=on ipv4
# ipadm set-ifprop -p forwarding=off -m ipv4 net0
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY              PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv4  forwarding            rw   on      on         off     on,off

# show-ifprop -p forwarding -m ipv4 net0
IFNAME PROPERTY        PROTO PERM CURRENT PERSISTENT DEFAULT  POSSIBLE
net0   forwarding      ipv4  rw   off     off        off      on,off
```

例 10 set-prop サブコマンドの修飾子の使用

次の一連のコマンドは、プラス修飾子とマイナス修飾子 (+、–) を使用して、1047、1048、および 1049 を TCP の追加の特権ポートとして追加します。

例 10 set-prop サブコマンドの修飾子の使用 (続き)

```
# ipadm set-prop -p extra_priv_ports=1047 tcp
# ipadm set-prop -p extra_priv_ports+=1048 tcp
# ipadm set-prop -p extra_priv_ports+=1049 tcp
# ipadm set-prop -p extra_priv_ports+=1050 tcp
```

次のコマンドは、1048 を追加の特権ポートとして削除します。

```
# ipadm set-prop -p extra_priv_ports-=1048
```

次のコマンドは、TCP の追加の特権ポートをすべて表示します。

```
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY             PERM CURRENT    PERSISTENT  DEFAULT   POSSIBLE
ipv4  extra_priv_ports     rw   1047,1049, 1047,1049,  2049,4045 1-65535
                                1050       1050
```

例 11 オブジェクトの有効化と無効化

次の一連のコマンドは、インタフェースオブジェクトとアドレスオブジェクトを有効または無効にし、それらの操作の結果を表示します。

```
# ipadm create-ip net1
# ipadm create-addr -T static -a local=10.2.3.4/24 net1/v4static
# ipadm set-addrprop -p private=yes net1/v4static
# ipadm show-addr net1/v4static
ADDROBJ        TYPE   STATE     ADDR
net1/v4static  static ok        10.2.3.4/24
```

次のコマンドは、アドレスオブジェクト `net1/v4static` を無効にします。

```
# ipadm disable-addr -t net1/v4static
# ipadm show-addr net1/v4static
ADDROBJ        TYPE   STATE     ADDR
net1/v4static  static ok        10.2.3.4/24
```

次のコマンドは、インタフェースオブジェクト `net1` を無効にします。

```
# ipadm disable-if -t net1
# ipadm show-if net1 -o all
IFNAME     CLASS    STATE    ACTIVE CURRENT       PERSISTENT OVER
net1       ip       disabled no     bm----------  --46       --
```

次のコマンドは、永続的な構成を使用してインタフェースオブジェクトを有効にします。

```
# ipadm enable-if -t net1
# ipadm show-if net1 -o all
IFNAME     CLASS    STATE    ACTIVE CURRENT       PERSISTENT OVER
net1       ip       ok       yes    bm--------46  --46       --
```

例 11 オブジェクトの有効化と無効化 *(続き)*

```
# ipadm show-addr net1/v4static
ADDROBJ        TYPE   STATE     ADDR
net1/v4static  static ok        10.2.3.4/24
```

インタフェースオブジェクトを有効にすると、そのインタフェースに構成されているすべてのアドレスオブジェクトも有効になります。

次のコマンドは、非大域排他的 IP ゾーンの net0 インタフェースの永続的な構成を作成します。これにより、次回のリブート時に大域ゾーンから allowed-address リソースを通して提供される一連のアドレスを使用して、net0 インタフェースが構成されます。

```
# ipadm create-ip net0
```

次のコマンドを実行すると、リブートしなくても、非大域排他的 IP ゾーンで使用可能な一連の allowed-address 値を使用して net0 インタフェースを構成できます。

```
# ipadm disable-if -t net0
# ipadm enable-if -t net0
```

例 12 IPMP インタフェースの作成

次の一連のコマンドは、IPMP インタフェースを作成し、基になるインタフェースを追加します。

```
# ipadm create-ip e1000g0
# ipadm create-ip e1000g1
# ipadm create-ip e1000g2
# ipadm set-ifprop -p standby=on -m ip e1000g2
# ipadm create-ipmp testgroup0
# ipadm add-ipmp -i e1000g0 -i e1000g1 -i e1000g2 testgroup0
# ipadm create-addr -T static -a local=192.168.80.5/24 testgroup0/data1
# ipadm create-addr -T static -a local=192.168.80.6/24 testgroup0/data2
# ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
net0       ip       ok       yes    --
e1000g0    ip       ok       yes    --
e1000g1    ip       ok       yes    --
ipmp0      ipmp     ok       yes    e1000g0 e1000g1
```

次の一連のコマンドは、IPMP インタフェースを無効にしてから有効にします。

```
# ipadm disable-if -t testgroup0
ipadm show-if
IFNAME     CLASS    STATE    ACTIVE OVER
lo0        loopback ok       yes    --
```

例 12 IPMP インタフェースの作成 (続き)

```
net0       ip       ok       yes    --
e1000g0    ip       disabled no     --
e1000g1    ip       disabled no     --
ipmp0      ipmp     disabled no     e1000g0 e1000g1
# ipadm enable-if -t testgroup0
```

次の一連のコマンドは、IPMP インタフェースから基になるインタフェースを削除したあとで、IPMP インタフェースを削除します。

```
ipadm remove-ipmp -i e1000g0 -i e1000g1 testgroup0
ipadm delete-ipmp testgroup0
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

read(1), arp(1M), cfgadm(1M), dhcpagent(1M), dladm(1M), if_mpadm(1M), ifconfig(1M), in.ndpd(1M), in.mpathd(1M), ndd(1M), zonecfg(1M), nsswitch.conf(4), attributes(5), dhcp(5), vni(7d)

『Oracle Solaris Tunable Parameters Reference Manual』

『RFC 791, Internet Protocol - DARPA Internet Program Protocol Specification』、Postel, J. (Information Sciences Institute, University of Southern California) 著、1981 年 9 月

『IP Version 6 Addressing Architecture』(RFC 4291)、Hinden, R. および S. Deering 著、2006 年 2 月

『IPv6 Stateless Address AutoConfiguration』(RFC 4862)、Thomson, S.、Narten, T.、および T. Jinmei 著、2007 年 9 月

『Dynamic Host Configuration Protocol for IPv6 (DHCPv6)』(RFC 3315)、Droms, R.、Bound, J.、Volz, B.、Lemon, T.、Perkins, C.、および M. Carney 著、2003 年 7 月

『Privacy Extensions for Stateless Address AutoConfiguration in IPv6』(RFC 4941)、Narten, T.、Draves, R.、および S. Krishnan 著、2007 年 9 月

『Management Information Base for the Internet Protocol (IP)』(RFC 4293)、S. Routhier, Ed. 著、2006 年 4 月

『Requirements for Internet Hosts - Communication Layers』(RFC 1122)、Braden, R. (Information Sciences Institute, University of Southern California) 著、1989 年 10 月

名前

ipf – IP パケットの入出力に関するパケットフィルタ処理リストの変更

形式

```
ipf [-6AdDEInoPRrsvVyzZ] [-l block | pass | nomatch]
     [-T optionlist] [-F i | o | a | s | S] -f filename
     [-f filename...]
```

機能説明

ipf ユーティリティーは、Solaris IP フィルタ機能に関連するコマンド群の 1 つです。ipfilter(5) を参照してください。

ipf ユーティリティーは、設定されたファイル名 (ハイフン (-) は標準入力として扱う) を開き、そのファイルを解析して、パケットフィルタの規則セットに対して追加または削除する規則のセットを抽出します。

解析に問題がない場合は、ipf によって処理された各規則がカーネルの内部リストに追加されます。規則は、ipf に指定したときの順序で内部リストの末尾に追加されます。

ipf の使用は、/dev/ipauth、/dev/ipl、および /dev/ipstate へのアクセスによって制限されます。これらのファイルのデフォルトのアクセス権では、ipf のすべての操作を root ユーザーとして実行する必要があります。

Solaris IP フィルタ機能の有効化

Solaris IP フィルタは、Solaris オペレーティングシステムとともにインストールされます。ただし、パケットフィルタ処理はデフォルトでは有効になっていません。次の手順で Solaris IP フィルタ機能をアクティブ化してください。

1. IP Filter Management の権利プロファイル (rbac(5) を参照) を持つ役割またはスーパーユーザーになります。
2. システムおよびサービスのファイアウォールポリシーを構成します。svc.ipfd(1M) および ipf(4) を参照してください。
3. (省略可能) ネットワークアドレス変換 (NAT) 構成ファイルを作成します。ipnat.conf(4) を参照してください。
4. (省略可能) アドレスプール構成ファイルを作成します。ippool(4) を参照してください。

   ひとかたまりのアドレスを単一のアドレスプールとして参照する場合は、ipool.conf ファイルを作成します。ブート時にアドレスプール構成ファイルをロードする場合は、アドレスプールを保存する /etc/ipf/ippool.conf というファイルを作成します。ブート時にアドレスプール構成ファイルをロードしない場合は、/etc/ipf 以外の場所に ippool.conf ファイルを保存し、規則を手動でアクティブ化します。
5. 次のようにして Solaris IP フィルタを有効にします。

   ```
   # svcadm enable network/ipfilter
   ```

パケットフィルタ処理を一時的に無効にしたあとで再度有効にするには、マシンをリブートするか、次のコマンドを入力します。

```
# svcadm enable network/ipfilter
```

これによって基本的に次の ipf コマンドを実行します。

1. Solaris IP フィルタを有効にします。

   # ipf -E

2. ippools を読み込みます。

   # ippool -f *<ippool configuration file>*

   ippool(1M) を参照してください。

3. (省略可能) パケットフィルタ処理をアクティブ化します。

   ipf -f *<ipf configuration file>*

4. (省略可能) NAT をアクティブ化します。

   ipnat -f *<IPNAT configuration file>*

   ipnat(1M) を参照してください。

注 – システムをリブートすると、IP フィルタ構成が自動的にアクティブ化されます。

オプション

次のオプションがサポートされています。

-6
このオプションは、IPv6 規則を解析する場合と読み込む場合に必要です。IPv6 規則の読み込みについては、将来変更される可能性があります。

-A
アクティブなリストを変更するためのリストを設定します (デフォルト)。

-d
デバッグモードを有効にします。フィルタ規則が処理されるたびに、生成されるフィルタ規則の 16 進ダンプが出力されます。

-D
フィルタを無効にします (有効になっている場合)。読み込み可能なカーネルのバージョンでは機能しません。

-E
フィルタを有効にします (無効になっている場合)。読み込み可能なカーネルのバージョンでは機能しません。

-F i|o|a
フラッシュするフィルタリストを指定します。パラメータとして、i (input、入力)、o (output、出力)、または a (all、すべてのフィルタ規則の削除) のいずれかを指定します。1 文字だけ指定することも、その文字で始まる語全体を指定することもできます。このオプションはほかのオプションの前またはあとに指定できますが、各オプションはコマンド行に指定された順序で実行されます。

-F s | S
状態テーブルからエントリをフラッシュするには、-F オプションと s (完全に確立されていない接続に関する状態情報を削除する場合) または S (状態テーブル全体を削除する場合) を組み合わせて使用します。これら 2 つのオプションは、どちらか 1 つしか指定できません。ipfstat -s の出力には、接続が完全に確立しなかったことを示す両方向の異常とともに、完全に確立された接続が 4/4 として表示されます。

-f *filename*
ipf がパケットフィルタの規則リストを変更するのに必要な入力の取得元ファイルを指定します。

-I
アクティブでないリストを変更するためのリストを設定します。

-l pass | block | nomatch
デフォルトのパケットログ機能を切り替えます。このオプションに対する有効な引数は、pass、block、および nomatch です。オプションを設定すると、フィルタ処理を終了し、設定カテゴリと一致したパケットがログに記録されます。これは、読み込まれたどの規則にも一致しないすべてのパケットをログに記録するのに非常に便利です。

-n
ipf が ioctl コールや、現在実行中のカーネルを変更する操作を実行しないようにします。

-o
デフォルトで、入力リスト (デフォルト) ではなく出力リストに規則を追加します (または出力リストから規則を削除します)。

-P
認証規則テーブルに一時的なエントリとして規則を追加します。

-R
IP アドレスからホスト名への解決とポート番号からサービス名への解決の両方を無効にします。

-r
一致するフィルタ規則を、内部リストに追加するのではなく、内部リストから削除します。

-s
現在アクティブなフィルタリストを代替リストと入れ替えます。

-T *optionlist*
IP フィルタのカーネル変数を実行時に変更できるようにします。変数によっては、変更できるようにするために IP フィルタを無効な状態 (-D) にする必要がある場合とそうでない場合があります。*optionlist* パラメータは、チューニングコマンドをコンマで区切ったリストです。チューニングコマンドは次のいずれかです。

list
カーネルに含まれるすべての変数と、その最大値、最小値、および現在の値のリストを取得します。

1 つの変数名
変数の現在の値を取得します。

変数名と割り当て値
新しい値を設定します。

次に例を示します。

```
# Print out all IPFilter kernel tunable parameters
ipf -T list

# Display the current TCP idle timeout and then set it to 3600
ipf -D -T fr_tcpidletimeout,fr_tcpidletimeout=3600 -E

# Display current values for fr_pass and fr_chksrc, then set
# fr_chksrc to 1.
ipf -T fr_pass,fr_chksrc,fr_chksrc=1
```

-v
詳細モードを有効にします。規則の処理に関する情報が表示されます。

-V
バージョン情報を表示します。ipf バイナリにコンパイルされたバージョン情報が表示されます。このバイナリはカーネルコードから取得されます (実行中または存在する場合)。ipf バイナリがカーネル内にある場合は、その現在の状態に関する情報 (ログ機能がアクティブかどうかや、デフォルトのフィルタ処理など) が表示されます。

-y
IP フィルタによって管理されているカーネル内部のインタフェースリストを、現在のインタフェースステータスリストを使って手動で再同期します。

-z
入力ファイル内の各規則について、その統計情報を 0 にリセットし、0 にする前の統計情報を表示します。

-Z
カーネルに保持されているフィルタ処理関連のグローバル統計情報だけを 0 にリセットします。フラグメントや状態の統計情報には影響しません。

ファイル

/dev/ipauth
/dev/ipl
/dev/ipstate
IP フィルタ疑似デバイスへのリンク。

/etc/ipf/ipf.conf
ipf の起動時構成ファイルの場所。ipf(4) を参照してください。

`/usr/share/ipfilter/examples/`
IP フィルタの例が多数含まれています。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	network/ipfilter
インタフェースの安定性	確実

関連項目

`ipfstat(1M)`, `ipmon(1M)`, `ipnat(1M)`, `ippool(1M)`, `svcadm(1M)`, `svc.ipfd(1M)`, `ipf(4)`, `ipnat.conf(4)`, `ippool(4)`, `attributes(5)`, `ipfilter(5)`

『Oracle Solaris の管理: IP サービス』

診断

カーネル内部のパケットフィルタ処理リストに実際にコマンドを適用するには、コマンドを root ユーザーとして実行する必要があります。

名前

ipfs – saves and restores information for NAT and state tables

形式

```
ipfs [-nv] -l
ipfs [-nv] -u
ipfs [-nv] [-d dirname] -R
ipfs [-nv] [-d dirname] -W
ipfs [-nNSv] [-f filename] -r
ipfs [-nNSv] [-f filename] -w
ipfs [-nNSv] -f filename -i <if1>,<if2>
```

機能説明

The `ipfs` utility enables the saving of state information across reboots. Specifically, the utility allows state information created for NAT entries and rules using "keep state" to be locked (modification prevented) and then saved to disk. Then, after a reboot, that information is restored. The result of this state-saving is that connections are not interrupted.

オプション

The following options are supported:

`-d` Change the default directory used with `-R` and `-W` options for saving state information.

`-n` Do not take any action that would affect information stored in the kernel or on disk.

`-v` Provides a verbose description of `ipfs` activities.

`-N` Operate on NAT information.

`-S` Operate on filtering state information.

`-u` Unlock state tables in the kernel.

`-l` Lock state tables in the kernel.

`-r` Read information in from the specified file and load it into the kernel. This requires the state tables to have already been locked and does not change the lock once complete.

`-w` Write information out to the specified file and from the kernel. This requires the state tables to have already been locked and does not change the lock once complete.

`-R` Restores all saved state information, if any, from two files, `ipstate.ipf` and `ipnat.ipf`, stored in the `/var/db/ipf` directory. This directory can be changed with the `-d` option. The state tables are locked at the beginning of this operation and unlocked once complete.

`-W` Saves in-kernel state information, if any, out to two files, `ipstate.ipf` and `ipnat.ipf`, stored in the `/var/db/ipf` directory. This directory can be changed with the `-d` option. The state tables are locked at the beginning of this operation and unlocked once complete.

ファイル

- /var/db/ipf/ipstate.ipf
- /var/db/ipf/ipnat.ipf
- /dev/ipl
- /dev/ipstate
- /dev/ipnat

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ipfilter
Interface Stability	Committed

関連項目

ipf(1M), ipmon(1M), ipnat(1M), attributes(5)

診断

Arguably, the -W and -R operations should set the locking and, rather than undo it, restore it to what it was previously.

Fragment table information is currently not saved.

名前

ipfstat – パケットフィルタ統計およびフィルタリストに関する報告

形式

```
ipfstat [-6aACdfghIilnoRstv]

ipfstat [-C] [-D addrport] [-P protocol] [-S addrport]
      [-T refreshtime]
```

機能説明

ipfstat コマンドは、Solaris IP フィルタ機能に関連するコマンド群の 1 つです。ipfilter(5) を参照してください。

ipfstat コマンドは、シンボル _fr_flags、_frstats、_filterin、および _filterout を使用して /dev/kmem を調べます。このコマンドを実行して機能させるには、/dev/kmem とカーネル自体を読み取れることが必要です。

デフォルトの動作では、ipfstat は、カーネルがフィルタを通してパケットをやりとりする間に蓄積された統計を取得して表示します。

ipfstat の役割は、有効になっているフィルタがあればそれらをカーネルに出入りするパケットに適用し、その結果として収集された現在のカーネル統計を表示することです。これは、コマンド行パラメータが指定されていない場合のデフォルトの動作です。-i または -o が指定された場合、ipfstat は、現在インストールされカーネルで使用されているフィルタ規則のリストを取得して表示します。

ipfstat はカーネルのデバイスファイルを使用して情報を取得します。これらのファイルのデフォルトのアクセス権では、ipfstat のすべての操作を root ユーザーとして実行する必要があります。

ipfstat コマンドは kstat(3KSTAT) カーネル機能をサポートしています。このサポートのため、ipfstat の代わりに kstat(1M) を使用できます。例:

kstat -m ipf

ipfstat -t オプションを使用すると、ipfstat は top 形式の状態表示モードになります。このモードでは、Unix の top ユーティリティーがプロセステーブルを表示するのと同様の方法で、状態テーブルが表示されます。-C、-D、- P、-S、および -T コマンド行オプションを使用すると、表示する状態エントリを限定したり、表示の更新頻度を指定したりできます。

top 形式の状態表示モードでは、次のキーを使用すると、表示する情報を変更できます。

d　表示する情報を選択します。

l　画面を描画し直します。

q　プログラムを終了します。

s　ソート基準を切り替えます。

r　ソート基準を逆にします。

状態エントリのプロトコル番号、IP パケット数、バイト数、および生存期間で状態をソートできます。デフォルトでは、バイト数でソートされます。状態は降順にソートされますが、r キーを使用すると昇順にソートできます。

発信元、着信先、およびプロトコルの各フィルタおよび更新頻度は、対話形式では変更できません。これはコマンド行から行う必要があります。

正しく表示するには、画面には少なくとも 80 カラム必要です。ただし、ipfstat では画面幅は検査されません。

ソート基準とフィルタ基準に一致する、最初の *X*-5 エントリだけが表示されます (*X* は画面の行数)。それ以上のエントリを表示する方法はありません。

オプション

次のオプションがサポートされています。

-6
: 可能であれば、IPv6 のフィルタリストと状態を表示します。このオプションは将来変更される可能性があります。

-a
: アカウンティングフィルタリストを表示し、各規則に対してカウントされたバイト数を表示します。

-A
: パケット認証統計を表示します。

-C
: -t との組み合わせでのみ有効です。「クローズ」状態も上位に表示します。通常、CLOSE_WAIT プロトコル状態に達した TCP 接続は表示されません。このオプションを有効にすると、すべての状態エントリが表示されます。

-d
: データを表示する際にデバッグ出力を生成します。

-D *addrport*
: -t との組み合わせでのみ有効です。top 形式の状態表示に表示する対象を、着信先 IP アドレスおよびポートが *addrport* 引数に一致する状態エントリに限定します。*addrport* は *ipaddress* [,*port*] という形式で指定します。*ipaddress* と *port* には、数値または文字列 any (この順に、任意の IP アドレスおよび任意のポートを指定) を指定する必要があります。-D オプションを指定しない場合、デフォルトで -D any,any になります。

-f
: フラグメントの状態情報 (統計) および保持された状態情報 (カーネル内) があれば表示します。

-g
: 現在設定されているグループ (アクティブなグループとアクティブでないグループの両方) を表示します。

-h
: 各規則で記録された「ヒット」の回数を規則ごとに表示します。-i と組み合わせて使用します。

-i
: カーネルの IP 処理の入力側に使用されるフィルタリストを表示します。

-I	アクティブでないフィルタリストの詳細を取得するか、アクティブなフィルタリストの詳細を取得するかを切り替えます。-i と組み合わせて使用します。
-l	-s オプションとともに使用すると、アクティブな状態エントリのリストが表示されます (統計なし)。
-n	各規則の出力時にその規則番号を表示します。
-o	カーネルの IP 処理の出力側に使用されるフィルタリストを表示します。
-P *protocol*	-t との組み合わせでのみ有効です。top 形式の状態表示に表示する対象を、特定のプロトコルに一致する状態エントリに限定します。引数には、プロトコル名 (/etc/protocols で定義されているもの) またはプロトコル番号を指定できます。このオプションを指定しない場合、 すべてのプロトコルの状態エントリが指定されます。
-R	IP アドレスからホスト名への解決とポート番号からサービス名への解決の両方を無効にします。
-S *addrport*	-t との組み合わせでのみ有効です。top 形式の状態表示に表示する対象を、発信元 IP アドレスおよびポートが *addrport* 引数に一致する状態エントリに限定します。*addrport* は *ipaddress* [,*port*] という形式で指定します。*ipaddress* と *port* には、数値または文字列 any (この順に、任意の IP アドレスおよび任意のポートを指定) を指定する必要があります。-S オプションを指定しない場合、 デフォルトで -S any,any になります。
-s	パケット/フローの状態情報 (統計のみ) を表示します。
-T *refreshtime*	-t との組み合わせでのみ有効です。top 形式の状態表示を更新する頻度を指定します。refreshtime は更新間隔の秒数です。任意の正の整数を使用できます。デフォルトは 1 (最小の更新間隔) です。
-t	Unix ユーティリティー top がプロセステーブルを表示するのと同様の方法で、状態テーブルを表示します。状態はいくつかの方法でソートできます。
-v	詳細モードを有効にします。追加のデバッグ情報を表示します。

ファイル

- /dev/kmem
- /dev/ksyms
- /dev/ipl
- /dev/ipstate

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	network/ipfilter
インタフェースの安定性	確実

関連項目

ipf(1M), kstat(1M), kstat(3KSTAT), attributes(5), ipfilter(5)

『Oracle Solaris の管理: IP サービス』

名前

ipmgmtd – IP ネットワークインタフェースおよび TCP/IP 調整可能パラメータの管理デーモン

形式

```
/lib/inet/ipmgmtd
```

```
svc:/network/ip-interface-management:default
```

機能説明

ipmgmtd は、ネットワークの IP インタフェースおよび IP/TCP/UDP/SCTP/ICMP 調整可能パラメータに関する管理イベントを処理するシステムデーモンです。このデーモンは、サービス管理機能 (SMF) サービスインスタンスによって制御されます。

```
svc:/network/ip-interface-management:default
```

ipmgmtd デーモンは SMF サービスによって自動的に起動されるので、直接起動しないようにしてください。これは、管理インタフェースやプログラミングインタフェースの構成要素ではありません。IP インタフェースおよび前述のプロトコルの調整可能パラメータを管理するための管理インタフェースは、ipadm(1M) によって提供されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	非公開

関連項目

ipadm(1M), attributes(5)

名前

ipmon – monitors /dev/ipl for logged packets

形式

```
ipmon [-abDFhnpstvxX] [-N device] [ [o] [NSI]] [-O [NSI]]
      [-P pidfile] [-S device] [-f device] [filename]
```

機能説明

The `ipmon` command is part of a suite of commands associated with the Solaris IP Filter feature. See `ipfilter(5)`.

The `ipmon` command opens `/dev/ipl` for reading and awaits data to be saved from the packet filter. The binary data read from the device is reprinted in human readable form. However, IP addresses are not mapped back to hostnames, nor are ports mapped back to service names. The output goes to standard output, by default, or a filename, if specified on the command line. Should the `-s` option be used, output is sent instead to `syslogd(1M)`. Messages sent by means of `syslog` have the day, month, and year removed from the message, but the time (including microseconds), as recorded in the log, is still included.

Messages generated by `ipmon` consist of whitespace-separated fields. Fields common to all messages are:

- The date of packet receipt. This is suppressed when the message is sent to `syslog`.
- The time of packet receipt. This is in the form *HH*:*MM*:*SS*.*F*, for hours, minutes, seconds, and fractions of a second (which can be several digits long).
- The name of the interface on which the packet was processed, for example, `ib1`.
- The group and rule number of the rule, for example, `@0:17`. These can be viewed with `ipfstat -in` for input rules or `ipfstat -in` for output rules. See `ipfstat(1M)`.
- The action: `p` for passed, `b` for blocked, `s` for a short packet, `n` did not match any rules, or `L` for a log rule.
- The addresses. This is actually three fields: the source address and port (separated by a comma), the symbol →, and the destination address and port. For example: `209.53.17.22,80 → 198.73.220.17,1722`.
- `PR` followed by the protocol name or number, for example, `PR tcp`.
- `len` followed by the header length and total length of the packet, for example, `len 20 40`.

If the packet is a TCP packet, there will be an additional field starting with a hyphen followed by letters corresponding to any flags that were set. See `ipf.conf(4)` for a list of letters and their flags.

If the packet is an ICMP packet, there will be two fields at the end, the first always being `icmp`, the next being the ICMP message and submessage type, separated by a slash. For example, `icmp 3/3` for a port unreachable message.

オプション

The following options are supported:

`-a`

Open all of the device logfiles for reading log entries. All entries are displayed to the same output device (stderr or syslog).

`-b`
For rules which log the body of a packet, generate hex output representing the packet contents after the headers.

`-D`
Cause `ipmon` to turn itself into a daemon. Using subshells or backgrounding of `ipmon` is not required to turn it into an orphan so it can run indefinitely.

`-f` *device*
Specify an alternative device/file from which to read the log information for normal IP Filter log records.

`-F`
Flush the current packet log buffer. The number of bytes flushed is displayed, even if the result is zero.

`-h`
Displays usage information.

`-n`
IP addresses and port numbers will be mapped, where possible, back into hostnames and service names.

`-N` *device*
Set the logfile to be opened for reading NAT log records from or to *device*.

`-o` *letter*
Specify which log files from which to actually read data. `N`, NAT logfile; `S`, state logfile; `I`, normal IP Filter logfile. The `-a` option is equivalent to using `-o NSI`.

`-O` *letter*
Specify which log files you do not wish to read from. This is most commonly used in conjunction with the `-a`. Letters available as parameters are the same as for `-o`.

`-p`
Cause the port number in log messages always to be printed as a number and never attempt to look it up.

`-P` *pidfile*
Write the PD of the `ipmon` process to a file. By default this is `/var/run/ipmon.pid`.

`-s`
Packet information read in will be sent through `syslogd` rather than saved to a file. The default facility when compiled and installed is `local0`. The following levels are used:

`LOG_INFO`
Packets logged using the `log` keyword as the action rather than `pass` or `block`.

`LOG_NOTICE`
Packets logged that are also passed.

LOG_WARNING
Packets logged that are also blocked.

LOG_ERR
Packets that have been logged and that can be considered “short”.

-S *device*
Set the logfile to be opened for reading state log records from or to *device*.

-t
Read the input file/device in the way performed by tail(1).

-v
Show TCP window, ack, and sequence fields

-x
Show the packet data in hex.

-X
Show the log header record data in hex.

ファイル

- /dev/ipl
- /dev/ipnat
- /dev/ipstate

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ipfilter
Interface Stability	Committed

関連項目

ipf(1M), ipfstat(1M), ipnat(1M), attributes(5), ipfilter(5)

『Oracle Solaris の管理: IP サービス』

診断

ipmon expects data that it reads to be consistent with how it should be saved and aborts if it fails an assertion which detects an anomaly in the recorded data.

名前

ipmpstat – IPMP サブシステムのステータスの表示

形式

```
ipmpstat [-n] [-o field[,...] [-P]] -a|-g|-i|-p|-t
```

機能説明

ipmpstat コマンドは、IPMP サブシステムに関する情報を簡潔に表示します。5 種類の出力モードがサポートされており、後述のとおり、IPMP サブシステムの状態をそれぞれ異なる視点で (アドレス、グループ、インタフェース、プローブ、およびターゲット) 表示できます。呼び出しごとに最大 1 つの出力モードを指定できます。表示される情報には自己矛盾のないことが保証されます。解析可能な出力形式も用意されているため、それをスクリプトで使用して IPMP サブシステムの状態を調べることができます。ipmpstat は基本的な特権のみで呼び出すことができますが、プローブモードだけはすべての特権を必要とします。

オプション

次のオプションがサポートされています。

-a
: IPMP データアドレスの情報を表示します (「アドレス」出力モード)。

-g
: IPMP グループの情報を表示します (「グループ」出力モード)。

-i
: IP インタフェースの情報を表示します (「インタフェース」出力モード)。

-n
: IP アドレスからホスト名への解決を試みずに、IP アドレスを数値で表示します。このオプションはどの出力モードでも使用できます。

-o *field*[,...]
: 指定された出力フィールドだけを、指定どおりの順序で表示します。フィールド名はコンマで区切って指定します。大文字と小文字が区別されます。サポートされるフィールド名は、選択した出力モードによって異なります (後述の説明を参照)。特殊なフィールド名 all を使用すると、特定の出力モードの全フィールドを表示できます。

-p
: IPMP プローブの情報を表示します (「プローブ」出力モード)。

-t
: IPMP ターゲットの情報を表示します (「ターゲット」出力モード)。

-P
: 後述のマシン解析可能な形式を使用して表示します。このオプションを指定する場合は、-o オプションを使用してフィールドのリストを明示的に指定する必要があります。

出力モード

ipmpstat ユーティリティーでは、次の出力モードがサポートされています。これらのモードは、前述のオプションの一部に対応しています。

アドレスモード

アドレスモードでは、システム上にあるすべての IPMP データアドレスの状態が表示されます。サポートされている出力フィールドは次のとおりです。

ADDRESS

情報に関連付けられているホスト名 (または IP アドレス)。「down」状態のアドレスには重複するものが存在する可能性があるため、一意の識別情報を形成するにはアドレスと GROUP を組み合わせる必要があります。ある特定の IPMP グループで重複アドレスが存在する場合は、最大 1 つが表示されます。また、「up」状態のアドレスが常に優先されます。

STATE

アドレスの状態。アドレスが IFF_UP (ifconfig(1M) を参照) であれば up、アドレスが IFF_UP でなければ down になります。

GROUP

このアドレスが属している IPMP IP インタフェース。

INBOUND

このアドレス宛のパケットを受信する、基になる IP インタフェース。これは、IP インタフェース障害などの外部イベントに応じて変更されることがあります。このフィールドが空の場合、システムはこのアドレス宛に送信された IP パケットを受け入れません (アドレスが「down」状態である場合や、その IPMP グループにはアクティブな IP インタフェースが残っていない場合など)。

OUTBOUND

この発信元アドレスを使用してパケットを送信する、基になる IP インタフェース。これは、IP インタフェース障害などの外部イベントに応じて変更されることがあります。このフィールドが空の場合、システムはこのアドレスを発信元としてパケットを送信することはありません (アドレスが「down」状態である場合や、その IPMP グループにはアクティブな IP インタフェースが残っていない場合など)。

-o を指定しない場合、 すべての出力フィールドが表示されます。

グループモード

グループモードでは、システム上にあるすべての IPMP グループの状態が表示されます。サポートされている出力フィールドは次のとおりです。

GROUP

情報に関連付けられている IPMP IP インタフェース名。匿名グループ (in.mpathd(1M) を参照) の場合、このフィールドは空になります。

GROUPNAME

IPMP グループ名。匿名グループの場合、このフィールドは空になります。

STATE

グループの状態。次のようになります。

ok　グループのすべてのインタフェースが使用可能です。

degraded　グループのインタフェースのすべてではなく一部のみが使用可能です。

failed　グループのすべてのインタフェースが使用不能です。

FDT
プローブベースの障害検出時間。プローブベースの障害検出が無効になっている場合、このフィールドは空になります。

INTERFACES
グループ内の基になる IP インタフェースのリスト。このリストは、3 つのパートに分かれています。

1. リストの最初の部分には、アクティブなインタフェースが表示されます。各括弧や丸括弧には囲まれません。アクティブなインタフェースとは、システムでデータトラフィックの送受信に使用されているインタフェースのことです。
2. 次に、INACTIVE 状態のインタフェースが丸括弧に囲まれて表示されます。INACTIVE 状態のインタフェースとは、機能してはいるが、管理ポリシーに従って使用されていないインタフェースのことです。
3. 最後に、使用不能なインタフェースが角括弧に囲まれて表示されます。使用不能なインタフェースとは、現在の設定ではまったく使用できないインタフェースのことです (FAILED や OFFLINE など)。

-o を指定しない場合、 すべての出力フィールドが表示されます。

インタフェースモード
インタフェースモードでは、システム上にあり in.mpathd によって追跡されているすべての IPMP インタフェースの状態が表示されます。サポートされている出力フィールドは次のとおりです。

INTERFACE
情報に関連付けられている IP インタフェース名。

ACTIVE
その IP インタフェースがシステムで IP データトラフィックに使用されているかどうかによって、yes または no になります。

GROUP
IP インタフェースに関連付けられている IPMP IP インタフェース。匿名グループ (in.mpathd(1M) を参照) の IP インタフェースの場合、このフィールドは空になります。

FLAGS
IP インタフェースに関するさまざまな情報。

i　INACTIVE 状態なので使用不能です。

s　STANDBY としてマークされています。

m　その IPMP グループの IPv4 マルチキャスト送受信用として指定されています。

b　その IPMP グループの IPv4 ブロードキャスト送受信用として指定されています。

M　その IPMP グループの IPv6 マルチキャスト送受信用として指定されています。

d　down 状態なので使用不能です。

h　ハードウェアアドレスの重複のため in.mpathd によって OFFLINE にされているので使用不能です。

LINK
リンクベースの障害検出の状態。

up
リンクは稼働しています。

down
リンクは停止しています。

unknown
ネットワークドライバからリンクの状態変化が報告されません。

PROBE
プローブベースの障害検出の状態。

ok
プローブで問題は検出されていません。

failed
プローブで障害が検出されました。

unknown
適切なプローブターゲットが認識されていないため、プローブを送信できません。

disabled
一意の IP テストアドレスが設定されていないため、プローブは無効になっています。

STATE
インタフェースの全体的な状態。

ok
設定されている障害検出方法によれば、インタフェースはオンライン状態で正しく動作しています。

failed
インタフェースはオンライン状態ですが、リンク状態が down またはプローブ状態が failed です。

offline
インタフェースはオフライン状態です。

unknown
インタフェースはオンライン状態ですが、設定されている障害検出方法が unknown 状態なので、動作しているかどうかは不明です。

-o を指定しない場合、 すべての出力フィールドが表示されます。

プローブモード
プローブモードでは、in.mpathd によって送信されているプローブに関する情報が表示されます。ほかの出力モードとは異なり、このモードは Ctrl-C で明示的に終了されるまで実行し続けます。サポートされている出力フィールドは次のとおりです。

TIME
ipmpstat の開始時からの時間で表した、プローブが送信された時間。ipmpstat の開始前にプローブが送信された場合、この時間は負になります。

PROBE
プローブを表す識別子。この識別子には、プローブタイプを示す接頭辞に続き、プローブの数値識別子が埋め込まれています。プローブタイプに使用できる値は次のとおりです。

i ICMP プローブ

t 推移的プローブ

ICMP プローブはアクティブなインタフェースから送信されます。プローブの数値識別子は、ある特定のアクティブインタフェースから in.mpathd によって IP プローブが送信されるたびに増分されます。数値識別子は ICMP プローブパケットの icmp_seq フィールドに一致します。これをパケット監視ツールで使用することで、より詳細な分析を行うことができます。

IPMP グループ内のどのインタフェースにも NOFAILOVER テストアドレスが設定されていない場合は、データパケットの受信にアクティブには使用されていないすべてのインタフェースから、推移的プローブが送信されます。推移的プローブの数値識別子は、ある特定のインタフェースから推移的プローブが送信されるたびに増分されます。プローブパケットの形式は実装内部のものです。

INTERFACE
プローブが送信された IP インタフェース。

TARGET
アクティブなインタフェースから送信された ICMP プローブのターゲットのホスト名 (または IP アドレス)。あるいは、推移的プローブが送信された IP インタフェースの名前。

NETRTT
プローブのネットワーク往復時間。これは、IP モジュールがプローブを送信してから肯定応答を受信するまでの時間です。プローブが失われたと in.mpathd で判断された場合、このフィールドは空になります。

RTT
プローブの合計往復時間。これは、in.mpathd がプローブ送信用のコードの実行を開始してから ack の処理を完了するまでの時間です。プローブが失われたと in.mpathd で判断された場合、このフィールドは空になります。合計往復時間が急上昇し、ネットワーク往復時間にはそのような急上昇が見られない場合は、ローカルシステム自体の過負荷を示しています。

RTTAVG
INTERFACE から TARGET までの往復時間の平均。これは、低速なターゲットの特定に役立ちます。十分なデータがないため平均を計算できない場合、このフィールドは空になります。

RTTDEV
INTERFACE から TARGET までの往復時間の標準偏差。これは、ばらつきの多いターゲットの特定に役立ちます。十分なデータがないため標準偏差を計算できない場合、このフィールドは空になります。

-o を指定しない場合、 RTTAVG と RTTDEV を除くすべてのフィールドが表示されます。

ターゲットモード
ターゲットモードでは、IPMP プローブターゲットの情報が表示されます。サポートされている出力フィールドは次のとおりです。

INTERFACE
情報に関連付けられている IP インタフェース名。

MODE
プローブターゲットの発見モード

routes	ルーティングテーブルによって見つかったプローブターゲット。
multicast	マルチキャスト ICMP プローブによって見つかったプローブターゲット。
disabled	プローブベースの障害検出はすべて無効になっています。
transitive	障害検出は推移的プローブによって行われます。このモードでは、グループ内にあるほかのアクティブインタフェースをプ

ローブで調べることにより、IP インタフェースの健全性が判定されます。

TESTADDR
: 送信プローブで使用される発信元アドレス。データトラフィックに使用されているアクティブなインタフェース、および NOFAILOVER テストアドレスが明示的に設定されているインタフェースについては、ICMP プローブの送受信に使用されるホスト名 (または IP アドレス) が表示されます。グループ内にあるほかのすべてのインタフェースについては、プローブの送信元のインタフェースの名前が表示されます。アクティブな IP インタフェースに IPv4 テストアドレスと IPv6 テストアドレスの両方が設定されている場合、プローブターゲットの情報は各テストアドレスについて個別に表示されます。

TARGETS
: ICMP プローブの場合はプローブターゲットのホスト名 (または IP アドレス) を、推移的プローブの場合はターゲットインタフェースを、それぞれスペースで区切ったリスト。IP ターゲットは調査順に表示されます。プローブターゲットが見つからなかった場合、このフィールドは空になります。

-o を指定しない場合、 すべての出力フィールドが表示されます。

出力形式

デフォルトでは、ipmpstat の出力モードには人間に読みやすい表形式が使用されます。各行には、特定の対象に関する情報が 1 つ以上のフィールドで表示されます。また、この対象は、これらのフィールドを 1 つ以上組み合わせることで一意に識別されます。この形式では、テーブルの上に (および 1 画面分の情報ごとに) フィールドを表すヘッダーが表示されます。フィールドはスペースで区切られ、空のフィールドは -- (2 つのハイフン) で表されます。ほかにも、読みやすくするための方法が使用されます。フィールドの値を特定できない場合、その値は「?」と表示され、標準エラーに診断メッセージが出力されます。

マシン解析可能な形式にも表形式が使用されますが、プログラムで効率よく解析できるように設計されています。具体的には、解析可能な形式は、人間に読みやすい形式と次の点で異なります。

- ヘッダーは表示されません。
- 空のフィールドには -- は表示されず、何も出力されません。
- フィールドの区切りは、可変な数のスペースではなく、1 つのコロン (:) です。
- 複数のフィールドが要求される場合、フィールドの値に文字としてコロン (:) またはバックスラッシュ (\) があると、これらの文字は前に \ を付けてエスケープされます。

使用例

例 1 特定のインタフェースの障害検出時間の取得

次のコードは、マシン解析可能な出力形式を使用して、特定の IPMP IP インタフェースの障害検出時間を出力する ksh 関数を作成します。

例 1 特定のインタフェースの障害検出時間の取得 (続き)

```
getfdt() {
    ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
        [[ "$group" = "$1" ]] && { echo "$fdt"; return; }
    done
}
```

属性

属性についての詳細は、attributes(5) を参照してください。

/usr/sbin/ipmpstat:

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実
マシン解析可能な形式	確実
人間に読みやすい形式	インタフェースではない

/usr/sbin/ipmpstat は「確実」インタフェースではありません。

関連項目

if_mpadm(1M), ifconfig(1M), in.mpathd(1M), attributes(5)

名前

ipnat – NAT サブシステムのユーザーインタフェース

形式

`ipnat [-CdFhlnRrsv] -f` *filename*

機能説明

`ipnat` ユーティリティーは、指定されたファイル (ハイフン (-) は標準入力として扱う) を開き、そのファイルを解析して、IP NAT に対して追加または削除する規則のセットを抽出します。

解析に問題がない場合は、`ipnat` によって処理された各規則がカーネルの内部リストに追加されます。規則は、`ipnat` に指定したときの順序で内部リストの末尾に追加されます。

`ipnat` の使用は、`/dev/ipauth`、`/dev/ipl`、および `/dev/ipstate` へのアクセスによって制限されます。これらのファイルのデフォルトのアクセス権では、`ipnat` のすべての操作を root ユーザーとして実行する必要があります。

`ipnat` の使用は、`/dev/ipnat` へのアクセスによって制限されます。`/dev/ipnat` のデフォルトのアクセス権では、`ipnat` のすべての操作を root ユーザーとして実行する必要があります。

オプション

次のオプションがサポートされています。

`-C`	現在の NAT 規則リストに含まれるすべてのエントリ (NAT 規則) を削除します。
`-d`	デバッグモードを有効にします。フィルタ規則が処理されるたびに、生成されるフィルタ規則の 16 進ダンプが出力されます。
`-F`	現在の NAT 変換テーブルに含まれるすべてのアクティブなエントリ (現在アクティブな NAT マッピング) を削除します。
`-f` *filename*	指定されたファイルを解析して、IP NAT に対して追加または削除する規則を抽出します。*filename* を標準入力にすることもできます。
`-h`	各マップ/リダイレクトフィルタのヒット数を出力します。
`-l`	現在の NAT テーブルエントリマッピングのリストを表示します。
`-n`	`ipf` が現在実行中のカーネルを変更する操作 (ioctl コールなど) を実行しないようにします。
`-R`	IP アドレスからホスト名への解決とポート番号からサービス名への解決の両方を無効にします。
`-r`	一致する NAT 規則を、内部リストに追加するのではなく、内部リストから削除します。
`-s`	NAT の統計情報を検索して表示します。
`-v`	詳細モードを有効にします。規則の処理およびアクティブな規則/テーブルエントリに関する情報が表示されます。

ファイル

/dev/ipnat	IP フィルタ疑似デバイスへのリンク。
/dev/kmem	仮想アドレス空間へのアクセスを提供する特殊ファイル。
/etc/ipf/ipnat.conf	ipnat の起動時構成ファイルの場所。
/usr/share/ipfilter/examples/	IP フィルタの例が多数含まれています。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	network/ipfilter
インタフェースの安定性	確実

関連項目

ipf(1M), ipfstat(1M), ipnat(4), attributes(5)

名前

ippool – user interface to the IP Filter pools

形式

```
ippool -a [-dnv] [-m poolname] [-o role] -i ipaddr
     [/netmask]

ippool -A [-dnv] [-m poolname] [-o role] [-S seed]
     [-t type]

ippool -f file [-dnuv]

ippool -F [-dv] [-o role] [-t type]

ippool -h [-dv] [-m poolname] [-t type]

ippool -l [-dv] [-m poolname] [-t type]

ippool -r [-dnv] [-m poolname] [-o role] -i ipaddr
     [/netmask]

ippool -R [-dnv] [-m poolname] [-o role] [-t type]

ippool -s [-dtv] [-M core] [-N namelist]
```

機能説明

The `ippool` utility is used to manage information stored in the IP pools subsystem of IP Filter software. Configuration file information can be parsed and loaded into the kernel and currently configured pools can be removed, changed, or inspected.

`ippool`'s use is restricted through access to `/dev/ippool`. The default permissions of `/dev/ippool` require `ippool` to be run as root for all operations.

The command line options used are divided into two sections: the global options and the instance-specific options.

`ippool`'s use is restricted through access to `/dev/ipauth`, `/dev/ipl`, and `/dev/ipstate`. The default permissions of these files require `ippool` to be run as root for all operations.

オプション

`ippool` supports the option categories described below.

Global Options

The following global options are supported:

- `-d` Toggle debugging of processing the configuration file.
- `-n` Prevents `ippool` from doing anything, such as making ioctl calls, that would alter the currently running kernel.
- `-v` Turn verbose mode on.

Instance-Specific Options

The following instance-specific options are supported:

- `-a` Add a new data node to an existing pool in the kernel.
- `-A` Add a new (empty) pool to the kernel.
- `-f` *file* Read in IP pool configuration information from *file* and load it into the kernel.
- `-F` Flush loaded pools from the kernel.

`-h`	Display a list of pools of the type: hash loaded in the kernel.
`-l`	Display a list of pools of the type: tree loaded in the kernel.
`-r`	Remove an existing data node from a pool in the kernel.
`-R`	Remove an existing pool from within the kernel.
`-s`	Display IP pool statistical information.

Other Options

The following, additional options are supported:

`-i` *ipaddr*[/*netmask*]	Sets the IP address for the operation being undertaken with an all-one's mask or, optionally, a specific netmask, given in either dotted-quad notation or as a single integer.
`-m` *poolname*	Sets the pool name for the current operation.
`-M` *core*	Specify an alternative path to `/dev/kmem` from which to retrieve statistical information.
`-N` *namelist*	Specify an alternative path to lookup symbol name information when retrieving statistical information.
`-o` *role*	Sets the role with which this pool is to be used. Currently only `ipf`, `auth`, and `count` are accepted as arguments to this option.
`-S` *seed*	Sets the hashing seed to the number specified. For use with hash-type pools only.
`-t` *type*	Sets the type of pool being defined. Must be one of `pool`, `hash`, or `group-map`.
`-u`	When parsing a configuration file, rather than load new pool data into the kernel, unload it.

ファイル

`/dev/ippool`	Link to IP Filter pseudo device.
`/dev/kmem`	Special file that provides access to virtual address space.
`/etc/ipf/ippool.conf`	Location of `ippool` startup configuration file.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ipfilter
Interface Stability	Committed

関連項目 ipf(1M), ipfstat(1M), ippool(4), attributes(5)

名前

ipqosconf – configure the IPQoS facility

形式

```
/usr/sbin/ipqosconf
/usr/sbin/ipqosconf -a conf_file [-vs]
/usr/sbin/ipqosconf -c
/usr/sbin/ipqosconf -f
/usr/sbin/ipqosconf -l
/usr/sbin/ipqosconf -L
```

機能説明

The `ipqosconf` utility configures the Quality of Service facility of the Internet Protocol (IP). Only superusers can use this command.

Without arguments, `ipqosconf` displays the actual `IPQoS` configuration.

Configuration is not preserved across reboot. You must apply the configuration every time that the machine reboots. To apply the configuration early in the boot phase, you can populate the `/etc/inet/ipqosinit.conf` file, which is then read from the `svc:/network/initial:default` service.

オプション

The following options are supported:

`-a` *conf_file*
: Apply the configuration in *conf_file*. If the *conf_file* is –, `ipqosconf` reads from standard input.

`-c`
: Populate the boot file with the current configuration.

`-f`
: Flush the configuration.

`-l`
: List the current applied configuration.

`-L`
: List the current configuration in verbose mode.

  In addition to the information that the `-l` option provides, the `-L` option provides filters and classes configured through other means than the `iqposconf` command. This option also provides the full set of filters that were created by `ipqosconf` by representing a multi-homed host in a configuration file

`-s`
: Log messages to `syslog` during an `-a` operation.

`-v`
: Toggle verbose mode during an `-a` operation.

  The `-v` option causes all messages to go to the console in addition to their normal destination. Messages intended to go to `syslog`, because the `-s` flag is set or because it is a log message, still go to `syslog` as well as the console.

Configuration File

The configuration file is composed of a format version and a succession of configuration (action) blocks. There are different configuration blocks for each type of action that is being configured.

Format Version

The first line of the configuration file specifies the format version contained in the configuration file.

The following entry specifies the format version:

```
fmt_version x.x
```

where *x.x* is the format version. `1.0` is the only supported version.

Configuration Blocks

Following the format version, are a succession of configuration (action) blocks that are different for each type of action being configured. A configuration block always has the following structure:

```
action {
      name action_name
      module module_name
      params_clause | ""
      cf_clauses
}

action_name      ::= string
module_name      ::= ipgpc | dlcosmk | dscpmk | flowacct | tswtclmt |
                     tokenmt

params_clause    ::= params {
                        parameters
                        params_stats | ""
                     }

parameters       ::= prm_name_value parameters | ""

prm_name_value   ::= param_name  param_value
```

Modules

The *param_name* and the types of *param_value* are specific to a given module.

```
params_stats     ::= global_stats boolean

cf_clauses       ::= class_clause cf_clauses |
                     filter_clause cf_clauses | ""

class_clause     ::= class {
                         name class_name
                         next_action next_action_name
                         class_stats | ""
                     }
```

```
class_name       ::= string
next_action_name ::= string
class_stats      ::= enable_stats boolean
boolean          ::= TRUE | FALSE

filter_clause    ::= filter {
                        name filter_name
                        class class_name
                        parameters
                     }

filter_name      ::= string
```

There must be exactly one configuration block belonging to module `ipgpc`. The action must be named `ipgpc.classify`. All other actions should be reachable from `ipgpc` by way of parameters of type action or the next_action of a class.

The set of types that are used for parameters of the different modules are:

```
action        ::=     string
protocol      ::=     1..255
port          ::=     1..65535
uint8         ::=     0..255
uint32        ::=     0..4294967296
int32         ::=     -2147483648..2147483648
address       ::=    <see the description section>
ifname        ::=    <interface name recognized by SIOGLIFINDEX ioctl>
enum          ::=     string | { string_list }
boolean       ::=     TRUE | FALSE
integer_array ::=     { range_value_list }
map_index     ::=     uint32
address       ::=     ip_address | ip_node_name
user          ::=     uid | username
uid           ::=     0..65535
username      ::=     string
string_list          ::=   string sl_entrys
sl_entrys            ::=   ',' string sl_entrys | ""
range_value_list     ::=   range_value_entry range_value_entrys
range_value_entry    ::=   range ':' integer_array_value
range                ::=   uint32 '-' uint32
integer_array_value  ::=   string | integer_array_number
integer_array_number ::=   uint8 | uint32
range_value_entrys   ::=   ';' range_value_entry range_value_entrys | ""
ip_node_name         ::=   string
ip_address           ::=   v4_address | v6_address
v4_address           ::=   v4_ip_address / v4_cidr_mask |
v4_ip_address
```

```
v4_cidr_mask         ::=   1-32
v6_address           ::=   v6_ip_address / v6_cidr_mask |
v6_ip_address
v6_cidr_mask         ::=   1-128
```

METER module tokenmt configuration syntax:

```
red_action_name         action
yellow_action_name      action
green_action_name       action
committed_rate          uint32
committed_burst         uint32
peak_rate               uint32
<if present this signifies that this will be a two rate meter, not
   a single rate meter>
peak_burst              uint32
<this is the 'peak' burst size for a two rate meter, but
   the 'excess' burst size for a single rate meter>
color_aware             boolean
color_map               integer_array
global_stats            boolean
```

METER module tswtclmt configuration syntax:

```
red_action_name         action
yellow_action_name      action
green_action_name       action
committed_rate          uint32
peak_rate               uint32
window                  uint32
global_stats            boolean
```

MARKER module dscpmk configuration syntax:

```
next_action         action
dscp_map            int_array
dscp_detailed_stats boolean
global_stats       boolean
```

MARKER module dlcosmk configuration syntax:

```
next_action         action
cos                 map_index
global_stats        boolean
```

CLASSIFIER module ipgpc configuration syntax:

```
user               user
projid             int32
if_name            ifname
```

```
direction          enum {
                   LOCAL_IN,
                   LOCAL_OUT,
                   FWD_IN,
                   FWD_OUT}
protocol           protocol
dsfield            uint8
dsfield_mask       uint8
saddr              address
daddr              address
sport              port
dport              port
priority           uint32
precedence         uint32
ip_version         enum {
                   V4,
                   V6 }
global_stats       boolean
```

ACCOUNTING module flowacct configuration syntax:

```
next_action      action
timer            uint32
timeout          uint32
max_limit        uint32
```

Types

action A string of characters with a matching action definition. The character string can be up to twenty three characters in length. To allow for spaces the string needs to be enclosed in quotes and cannot span lines. Two special actions are pre-defined and can not have an explicit action definition. The two pre-defined actions are `continue` and `drop`. continue causes the packet that is passed to it to continue normal processing. `drop` causes the packet that is passed to it to be dropped.

address A machine name or address recognized by `getipnodebyname(3SOCKET)`. If a machine name is specified, and `ip_version` has been defined, the query is done using that address family. If a machine name is not specified and `ip_version` has not been defined, the query is done using the `AI_DEFAULT` flag to `getipnodebyname()(..AF_INET6..)`. `CIDR` address masks following an IP address are allowed. Specify the `CIDR` address masks as `1-32` (for `v4`) or `1-128` (for `v6`). `CIDR` addresses are disallowed for node names.

enum Either one of the supported values or comma delimited list of support values, enclosed in curly braces.

ifname A non-`NULL`, existing interface name recognized by the `SIOGLIFINDEX` socket ioctl.

integer_array A comma delimited set of *range*/*value* pairs, enclosed in curly braces.

Specify *range* in the format *x-y*, where *x* and *y* are integers that denote the range of array indexes to which the value applies. The minimum value for both *x* and *y* is `0`. The maximum value for *x* is particular to the parameter. Any array indexes not referred to in the set of ranges are left at their previous value.

map_index A non-negative integer used as an index into any maps associated with a parameter of this type.

The maximum value of this type is dictated by the number of entries in the associated maps. The index starts at `0`.

port Either a service name recognized by `getservbyname(3SOCKET)` or an integer `1-65535`.

protocol Either a protocol name recognized by `getprotobyname(3SOCKET)` or an integer `1-255`.

string A character string. Enclose *string* in quotes. *string* cannot span multiple lines.

user Either a valid user ID or username for the system that is being configured.

Parameters

The configuration file can contain the following parameters

color_aware A value of `TRUE` or `FALSE`, indicating whether or not the configured action takes account of the previous packet coloring when classifying.

color_map An integer array that defines which values of the `dscp` field correspond with which colors for when the `color_aware` parameter is set to `TRUE`.

committed_burst The committed burst size in bits.

committed_rate The committed rate in bits per second.

cos The value used to determine the underlying driver level priority applied to the packet which is defined in `802.1D`.

daddr The destination address of the datagram.

direction The value used to build a filter matching only part of the traffic.

This parameter is of type `enum` with valid values of `LOCAL_IN` (local bound traffic), `LOCAL_OUT` (local sourced traffic), `FWD_IN` (forwarded traffic entering the system), and `FWD_OUT` (forwarded traffic exiting the system).

dport The destination port of the datagram.

dscp_detailed_stats	A value of `TRUE` or `FALSE` that determines whether detailed statistics are switched on for this `dscp` action. Specify `TRUE` to switch on or `FALSE` to switch off.
dscp_map	The *integer_array* that supplies the values that IP packets with a given `dscp` value have their dscp re-marked with. The existing value is used to index into the array where the new value is taken from. The array is of size `64`, meaning valid indexes are `0-63` and valid values are also `0-63`.
dsfield	The `DS` field of the `IP` datagram header. This is an 8–bit value, with each bit position corresponding with the same one in the header; this enables matches to be done on the CU bits. If you specify this parameter, you must also specify the `dsfield_mask` parameter.
`dsfield_mask`	The mask applied to the `dsfield` parameter to determine the bits against which to match. This is an 8-bit value, with each bit position corresponding with the same one in the `dsfield` parameter.
global_stats	A value of `TRUE` or `FALSE` to enable or disable the statistic collection for this action.
green_action_name	The action to be executed for packets that are deemed to be green.
if_name	The name of an interface recognized by the `SIOGLIFINDEX` ioctl. This parameter is of type `ifname`.
ip_version	This parameter is of type `enum` and has valid values of `V4` and `V6`. If it is set to `V4` only then only `ipv4`addresses are requested for a specified hostname. If it is set to `V6`, only `ipv6` addresses are returned if there are any, otherwise `v4` mapped `v6` addresses are returned. If both `V4` and `V6` are specified, or if `ip_version` is not specified, then both `ipv4` and `ipv6` addresses are requested for a specified hostname.
max_limit	The maximum number of flow entries present at one time in the `flowacct` actions in the memory resident table.
next_action	The action to be executed when the current action is complete. This value can be either the name of an action defined in the configuration file, or one of the two special action types: `drop` and `continue`.
peak_burst	The peak burst size, for a two rate meter, or excess burst size, for a single rate meter, in bits.
peak_rate	The peak rate in bits per second.

precedence	An integer that is used to order filters. If there are two matching filters that have the same priority value, the one with the lower precedence value is the one matched. This parameter should be used because the order of the filters in a configuration file has no influence on their relative precedence.
priority	An integer that represents the relative priority of a filter. If there are two matching filters, the one with the higher priority value is the one matched. Multiple filters can have the same priority.
projid	The project ID of the process sending the data. This value is always `-1` for received traffic.
protocol	The Upper Layer Protocol against which this entry is matched.
red_action_name	The action to be executed for packets that are determined to be red.
saddr	The source address of the datagram.
sport	The source port of the datagram.
timeout	The timeout in milliseconds after which flows are written to the accounting file.
timer	The period in milliseconds at which timed-out flows are checked for.
user	The user ID or username of the process sending the data. This value is always `-1` for received traffic.
window	The window size in ms.
yellow_action_name	The action to be executed for packets that are determined to be yellow.

セキュリティー

None.

使用例

例 1　Sending All Traffic From eng to the AF 1 Class of Service

This example sends all traffic from `eng` to the `AF 1` class of service. It is documented in four separate steps:

The following step creates a `tokenmt` action with three outcomes:

```
#meter for class 1.
action {
        name AF_CL1
        module tokenmt
        params{
                committed_rate 64
                committed_burst 75
                peak_burst 150
```

例 1　Sending All Traffic From eng to the AF 1 Class of Service　(続き)

```
                global_stats TRUE
                red_action_name drop
                yellow_action_name markAF12
                green_action_name markAF11
        }
}
```

The following step creates two dscpmk actions:

```
#class 1, low drop precedence.
action {
        name markAF11
        module dscpmk
        params{
             dscp_map {0-63:28}
             dscp_detailed_stats TRUE
                global_stats TRUE
                next_action acct1
        }
}
#class 1, medium drop precedence.
action {
        name markAF12
        module dscpmk
        params {
                dscp_map {0-63:30}
             dscp_detailed_stats TRUE
                global_stats TRUE
                next_action acct1
        }
}
```

The following step creates an accounting action:

```
#billing for transmitted class 1 traffic.
action {
        name acct1
        module flowacct
        params {
                timer 10
                timeout 30
                global_stats TRUE
max_limit 1024
next_action continue
        }
}
```

例 1 Sending All Traffic From eng to the AF 1 Class of Service (続き)

The following step creates an ipgpc action:

```
#traffic from eng sent, traffic from ebay dropped.
action {
        name ipgpc.classify
        module ipgpc
        class {
                name from_eng
                enable_stats TRUE
                next_action AF_CL1
        }
        class {
                name from_ebay
                enable_stats TRUE
                next_action drop
        }

        filter {
                name from_eng
                saddr eng-subnet
                class from_eng
        }
        filter {
                name from_ebay
                saddr ebay-subnet
                class from_ebay
        }
}
```

ファイル

/etc/inet/ipqosinit.conf	Contains the IPQoS configuration loaded at boot time. If this file exists, it is read from the network/initial:default service.
/etc/inet/ipqosconf.1.sample	Sample configuration file for an application server
/etc/inet/ipqosconf.2.sample	Sample configuration file that meters the traffic for a specified application
/etc/inet/ipqosconf.3.sample	Sample configuration file that marks the ethernet headers of web traffic with a given user priority

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ipqos

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

syslog(3C), getipnodebyname(3SOCKET), getprotobyname(3SOCKET), getservbyname(3SOCKET), attributes(5), dlcosmk(7ipp), dscpmk(7ipp), flowacct(7ipp), ipgpc(7ipp), ipqos(7ipp), tokenmt(7ipp), tswtclmt(7ipp)

診断

ipqosconf sends messages to syslog of facility user, severity notice when any changes are made to the IPQoS configuration.

Errors that occur during an ipqosconf operation send an error message to the console by default. For the application of a new configuration if the -s option is set then these messages are sent to syslog as facility user, severity error instead. If the -v option is present during an application then all error and change notificationmessages are sent to the console as well as their default destination.

名前

ipsecalgs – configure the IPsec protocols and algorithms table

形式

```
ipsecalgs

ipsecalgs -l

ipsecalgs -s

ipsecalgs -a [-P protocol-number | -p protocol-name] -k keylen-list
     [-i inc] [-K default-keylen] -b blocklen-list -n alg-names
     -N alg-number -m mech-name [-I initialization-vector_length]
     [-M MAC-length] [-S length-of-salt] [-F flags] [-f] [-s]

ipsecalgs -P protocol-number -p protocol-name
     [-e exec-mode] [-f] [-s]

ipsecalgs -r -p protocol-name [] -n alg-name [-s]

ipsecalgs -r -p protocol-name [] -N alg-number [-s]

ipsecalgs -R -P protocol-number [-s]

ipsecalgs -R -p protocol-name [-s]

ipsecalgs -e exec-mode -P protocol-number [-s]

ipsecalgs -e exec-mode -p protocol-name [-s]
```

機能説明

Use the `ipsecalgs` command to query and modify the IPsec protocol and algorithms stored in `/etc/inet/ipsecalgs`. You can use the `ipsecalgs` command to do the following:

- list the currently defined IPsec protocols and algorithms
- modify IPsec protocols definitions
- modify IPsec algorithms definitions

Never edit the `/etc/inet/ipsecalgs` file manually. The valid IPsec protocols and algorithms are described by the ISAKMP DOI. See RFC 2407. In the general sense, a Domain of Interpretation (DOI) defines data formats, network traffic exchange types, and conventions for naming security-relevant information such as security policies or cryptographic algorithms and modes. For `ipsecalgs`, the DOI defines naming and numbering conventions for algorithms and the protocols they belong to. These numbers are defined by the Internet Assigned Numbers Authority (IANA). Each algorithm belongs to a protocol. Algorithm information includes supported key lengths, block or MAC length, and the name of the cryptographic mechanism corresponding to that algorithm. This information is used by the IPsec modules, `ipsecesp(7P)` and `ipsecah(7P)`, to determine the authentication and encryption algorithms that can be applied to IPsec traffic.

The following protocols are predefined:

`IPSEC_PROTO_ESP`	Defines the encryption algorithms (transforms) that can be used by IPsec to provide data confidentiality.
`IPSEC_PROTO_AH`	Defines the authentication algorithms (transforms) that can be used by IPsec to provide authentication.

The mechanism name specified by an algorithm entry must correspond to a valid Solaris Cryptographic Framework mechanism. You can obtain the list of available mechanisms by using the cryptoadm(1M) command.

Applications can retrieve the supported algorithms and their associated protocols by using the functions getipsecalgbyname(3NSL), getipsecalgbynum(3NSL), getipsecprotobyname(3NSL) and getipsecprotobynum(3NSL).

Modifications to the protocols and algorithm by default update only the contents of the /etc/inet/ipsecalgs configuration file. In order for the new definitions to be used for IPsec processing, the changes must be communicated to the kernel using the -s option. See NOTES for a description of how the ipsecalgs configuration is synchronized with the kernel at system restart.

When invoked without arguments, ipsecalgs displays the list of mappings that are currently defined in /etc/inet/ipsecalgs. You can obtain the corresponding kernel table of protocols and algorithms by using the -l option.

オプション

ipsecalgs supports the following options:

-a
: Adds an algorithm of the protocol specified by the -P option. The algorithm name(s) are specified with the -n option. The supported key lengths and block sizes are specified with the -k, -i, and -b options.

-b
: Specifies the block or MAC lengths of an algorithm, in bytes. Set more than one block length by separating the values with commas.

-e
: Designates the execution mode of cryptographic requests for the specified protocol in the absence of cryptographic hardware provider. See cryptoadm(1M). *exec-mode* can be one of the following values:

  sync
  : Cryptographic requests are processed synchronously in the absence of a cryptographic hardware provider. This execution mode leads to better latency when no cryptographic hardware providers are available

  async
  : Cryptographic requests are always processed asynchronously in the absence of cryptographic hardware provider. This execution can improve the resource utilization on a multi-CPU system, but can lead to higher latency when no cryptographic hardware providers are available.

  This option can be specified when defining a new protocol or to modify the execution mode of an existing protocol. By default, the sync execution mode is used in the absence of a cryptographic hardware provider.

`-f`
Used with the `-a` option to force the addition of an algorithm or protocol if an entry with the same name or number already exists.

`-i`
Specifies the valid key length increments in bits. This option must be used when the valid key lengths for an algorithm are specified by a range with the `-k` option.

`-K`
Specifies the default key lengths for an algorithm, in bits. If the `-K` option is not specified, the minimum key length will be determined as follows:

- If the supported key lengths are specified by range, the default key length will be the minimum key length.
- If the supported key lengths are specified by enumeration, the default key length will be the first listed key length.

`-k`
Specifies the supported key lengths for an algorithm, in bits. You can designate the supported key lengths by enumeration or by range.

Without the `-i` option, `-k` specifies the supported key lengths by enumeration. In this case, *keylen-list* consists of a list of one or more key lengths separated by commas, for example:

```
128,192,256
```

The listed key lengths need not be increasing, and the first listed key length will be used as the default key length for that algorithm unless the `-K` option is used.

With the `-i` option, `-k` specifies the range of supported key lengths for the algorithm. The minimum and maximum key lengths must be separated by a dash ('`-`') character, for example:

```
32-448
```

`-l`
Displays the kernel algorithm tables.

`-m`
Specifies the name of the cryptographic framework mechanism name corresponding to the algorithm. Cryptographic framework mechanisms are described in the `cryptoadm(1M)` man page.

`-N`
Specifies an algorithm number. The algorithm number for a protocol must be unique. IANA manages the algorithm numbers. See RFC 2407.

`-n`
Specifies one or more names for an algorithm. When adding an algorithm with the `-a` option, *alg-names* contains a string or a comma-separated list of strings, for example:

`des-cbs,des`

When used with the `-r` option to remove an algorithm, *alg-names* contains one of the valid algorithm names.

`-P`

Adds a protocol of the number specified by *protocol-number* with the name specified by the `-p` option. This option is also used to specify an IPsec protocol when used with the `-a` and the `-R` options. Protocol numbers are managed by the IANA. See RFC 2407.

`-p`

Specifies the name of the IPsec protocol.

`-R`

Removes and IPsec protocol from the algorithm table. The protocol can be specified by number by using the `-P` option or by name by using the `-p` option. The algorithms associated with the protocol are removed as well.

`-r`

Removes the mapping for an algorithm The algorithm can be specified by algorithm number using the `-N` option or by algorithm name using the `-A` option.

`-s`

Synchronizes the kernel with the contents of `/etc/inet/ipsecalgs`. The contents of `/etc/inet/ipsecalgs` are always updated, but new information is not passed on to the kernel unless the `-s` is used. See `NOTES` for a description of how the `ipsecalgs` configuration is synchronized with the kernel at system restart.

The following options allow optional parameters to be configured. These are currently only used for combined mode algorithms, that is, algorithms that provide encryption and authentication in a single operation.

`-I`

The length of the Initialization Vector (IV) in bytes. The default IV length is the same as the block length.

`-M`

The length of the MAC or ICV in bytes for combined mode algorithms.

`-S`

The number of bytes of salt needed by the algorithm. The salt needs to be provided by the key management mechanism.

`-F`

Algorithm flags. These influence the way in which the kernel handles security tasks, especially authentication, in the kernel. They are also used by `ipseckey(1M)` and `ipsecconf(1M)`. Flags can be specified as a comma-separated list of tokens; see the example below. The following tokens are supported:

COUNTERMODE
: The algorithm uses counter mode.

COMBINED
: The algorithm provides encryption and authentication in the same operation.

CCM
: The cryptographic framework mechanism needs a CK_AES_CCM_PARAMS structure.

GMAC
: The cryptographic framework mechanism needs a CK_AES_GMAC_PARAMS structure.

GCM
: The cryptographic framework mechanism needs a CK_AES_GCM_PARAMS structure.

CBC
: This flag indicates the algorithm uses Cipher-block chaining. The cryptographic framework mechanism does not need a params structure. This is also the default, this flag can be omitted.

The algorithm flags can be displayed with the -l option.

使用例

例 1 Adding a Protocol for IPsec Encryption

The following example shows how to add a protocol for IPsec encryption:

```
example# ipsecalgs -P 3 -p "IPSEC_PROTO_ESP"
```

例 2 Adding the Blowfish Algorithm

The following example shows how to add the Blowfish algorithm:

```
example# ipsecalgs -a -P 3 -k 32-488 -K 128 -i 8 -n "blowfish" \
  -b 8 -N 7 -m CKM_BF_CBC
```

例 3 Updating the Kernel Algorithm Table

The following example updates the kernel algorithm table with the currently defined protocol and algorithm definitions:

```
example# svcadm refresh ipsecalgs
```

例 4 Adding the AES Galois/Counter Mode (GCM) Algorithm

The following command adds this algorithm.

```
example# ipsecalgs -a -P3 -k 128-256 -K 128 -i 64 -N 20 -b 16 \
-n "aes-gcm16,aes-gcm" -m CKM_AES_GCM -M 16 -I 8 -S 4 \
-F GCM,COMBINED,COUNTER
```

ファイル

`/etc/inet/ipsecalgs`

File that contains the configured IPsec protocols and algorithm definitions. Never edit this file manually.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`cryptoadm(1M)`, `ipsecconf(1M)`, `ipseckey(1M)`, `svcadm(1M)`, `getipsecalgbyname(3NSL)`, `getipsecprotobyname(3NSL)`, `ike.config(4)`, `attributes(5)`, `smf(5)`, `ipsecah(7P)`, `ipsecesp(7P)`

Piper, Derrell, RFC 2407, The Internet IP Security Domain of Interpretation for ISAKMP. Network Working Group. November 1998.

注意事項

When protocols or algorithm definitions that are removed or altered, services that rely upon these definitions can become unavailable. For example, if the `IPSEC_PROTO_ESP` protocol is removed, then IPsec cannot encrypt and decrypt packets.

Synchronization of the `ipsecalgs` configuration with the kernel at system startup is provided by the following `smf(5)` service:

```
svc:/network/ipsec/ipsecalgs:default
```

The IPsec services are delivered as follows:

```
svc:/network/ipsec/policy:default (enabled)
svc:/network/ipsec/ipsecalgs:default (enabled)
svc:/network/ipsec/manual-key:default (disabled)
svc:/network/ipsec/ike:default (disabled)
```

Services that are delivered disabled are delivered that way because the system administrator must create configuration files for those services before enabling them. See `ipseckey(1M)` and `ike.config(4)`. The default policy for the `policy` service is to allow all traffic to pass without IPsec protection. See `ipsecconf(1M)`.

The correct administrative procedure is to create the configuration file for each service, then enable each service using `svcadm(1M)`, as shown in the following example:

```
example# svcadm enable ipsecalgs
```

The service's status can be queried using the `svcs(1)` command.

If the `ipsecalgs` configuration is modified, the new configuration should be resynchronized as follows:

```
example# svcadm refresh ipsecalgs
```

Administrative actions on this service, such as enabling, disabling, refreshing, and requesting restart can be performed using svcadm(1M). A user who has been assigned the authorization shown below can perform these actions:

```
solaris.smf.manage.ipsec
```

See auths(1), user_attr(4), rbac(5).

The ipsecalgs smf(5) service does not have any user-configurable properties.

The smf(5) framework records any errors in the service-specific log file. Use any of the following commands to examine the logfile property:

```
example# svcs -l ipsecalgs
example# svcprop ipsecalgs
example# svccfg -s ipsecalgs listprop
```

This command requires sys_ip_config privilege to operate and thus can run in the global zone and in exclusive-IP zones. All shared-IP zones share the same available set of algorithms; however, you can use ipsecconf(1M) to set up system policy that uses differing algorithms for various shared-IP zones. All exclusive-IP zones have their own set of algorithms.

名前

ipsecconf – configure system wide IPsec policy

形式

```
/usr/sbin/ipsecconf

/usr/sbin/ipsecconf -a file [-q]

/usr/sbin/ipsecconf -c file

/usr/sbin/ipsecconf -d [-i tunnel-name] {index, tunnel-name, index}

/usr/sbin/ipsecconf -f [-i tunnel-name]

/usr/sbin/ipsecconf -F

/usr/sbin/ipsecconf -Fa file [-q]

/usr/sbin/ipsecconf -l [-i tunnel-name] [-n]

/usr/sbin/ipsecconf -L [-n]

/usr/sbin/ipsecconf -r file [-q]
```

機能説明

The `ipsecconf` utility configures the IPsec policy for a host or for one of its tunnels. Once the policy is configured, all outbound and inbound datagrams are subject to policy checks as they exit and enter the host or tunnel. For the host policy, if no entry is found, no policy checks will be completed, and all the traffic will pass through. For a tunnel, if no entry is found and there is at least one entry for the tunnel, the traffic will automatically drop. The difference in behavior is because of the assumptions about IPsec tunnels made in many implementations. Datagrams that are being forwarded will not be subjected to policy checks that are added using this command. See `ifconfig(1M)` and `dladm(1M)` for information on how to protect forwarded packets. Depending upon the match of the policy entry, a specific action will be taken.

This command can be run only by superuser.

Each entry can protect traffic in either one direction (requiring a pair of entries) or by a single policy entry which installs the needed symmetric `sadb` rules.

When the command is issued without any arguments, the list of file policy entries loaded are shown. To display the (`spd p.e.s`) use the `-l` option. Both will display the index number for the entry. To specify a single tunnel's SPD, use the `-i` option in combination with `-l`. To specify all SPDs, both host and for all tunnels, use `-L`.

Note, since one file policy entry (FPE) can generate multiple SPD pol entries (SPEs), the list of FPEs may not show all the actual entries. However, it is still useful in determining what what rules have been added to get the spd into its current state.

You can use the `-d` option with the index to delete a given policy in the system. If the `-d` option removes an FPE entry that produces multiple SPEs, only then SPD with the same policy index as the FPE will be removed. This can produce a situation where there may be SPEs when there are no FPEs.

As with `-l`, `-d` can use the `-i` flag to indicate a tunnel. An alternate syntax is to specify a tunnel name, followed by a comma (`,`), followed by an index. For example, `ip.tun0,1`.

With no options, the entries are displayed in the order that they were added, which is not necessarily the order in which the traffic match takes place.

To view the order in which the traffic match will take place, use the `-l` option. The rules are ordered such that all bypass rules are checked first, then ESP rules, then AH rules. After that, they are checked in the order entered.

Policy entries are not preserved across system restarts. Permanent policy entries should be added to `/etc/inet/ipsecinit.conf`. This file is read by the following `smf(5)` service:

```
svc:/network/ipsec/policy
```

See `NOTES` for more information on managing IPsec security policy and `SECURITY` for issues in securing `/etc/inet/ipsecinit.conf`.

オプション

`ipsecconf` supports the following options:

`-a` *file*

Add the IPsec policy to the system as specified by each entry in the file. An IPsec configuration file contains one or more entries that specify the configuration. Once the policy is added, all outbound and inbound datagrams are subject to policy checks.

Entries in the files are described in the オペランド section below. Examples can be found in the 使用例 section below.

Policy is latched for TCP/UDP sockets on which a `connect(3SOCKET)` or `accept(3SOCKET)` is issued. So, the addition of new policy entries may not affect such endpoints or sockets. However, the policy will be latched for a socket with an existing non-null policy. Thus, make sure that there are no preexisting connections that will be subject to checks by the new policy entries.

The feature of policy latching explained above may change in the future. It is not advisable to depend upon this feature.

The default behavior is to append new rules to the existing policy. If a new rule conflicts with an existing rule, an error is reported, the new rule will not be added.

To add a new rule that conflicts with an existing rule, the existing rule must be removed (see below) or the existing policy flushed. If the policy is flushed, IPsec will not protect any network traffic until a new policy is added.

The `-F` (flush) flag can be combined with `-a` to perform an atomic policy replacement; the existing policy will be replaced by the new policy described in the config file. By combining these two flags, there is not even a small window when the system is without a policy, which is the case when the flush and add commands are run sequentially.

`-c` *file*

Check the syntax of the configuration file and report any errors without making any changes to the policy. This option is useful when debugging configurations and when `smf(5)` reports a configuration error. See `SECURITY`.

`-d` *index*
: Delete the host policy denoted by the index. The index is obtained by invoking `ipsecconf` without any arguments, or with the `-l` option. See DESCRIPTION for more information. Once the entry is deleted, all outbound and inbound datagrams affected by this policy entry will not be subjected to policy checks. Be advised that with connections for which the policy has been latched, packets will continue to go out with the same policy, even if it has been deleted. It is advisable to use the `-l` option to find the correct policy index.

`-d` *name,index*
: Delete the policy entry denoted by *index* on a tunnel denoted by *name*. Since tunnels affect traffic that might originate off-node, latching does not apply as it does in the host policy case. Equivalent to: `-d` *index* `-i` *name*.

`-f`
: Flush all the policies in the system. Constraints are similar to the `-d` option with respect to latching and host versus per-tunnel behavior.

`-F`
: Flush all policies on all tunnels and also flush all host policies. See discussion of combining the `-F` and `-a` options, under `-a`, above.

`-i` *name*
: Specify a tunnel interface name for use with the `-d`, `-f`, or `-l` flags.

`-l`
: Listing of a single policy table, defaulting to the host policy. When `ipsecconf` is invoked without any arguments, a complete list of policy entries with indexes added by the user since boot is displayed. The current table can differ from the previous one if, for example, a multi-homed entry was added or policy reordering occurred, or if a single rule entry generates two `spd` rules In the case of a multi-homed entry, all the addresses are listed explicitly. If a mask was not specified earlier but was instead inferred from the address, it will be explicitly listed here. This option is used to view policy entries in the correct order. The outbound and inbound policy entries are listed separately.

`-L`
: Lists all policy tables, including host policy and all tunnel instances (including configured but unplumbed).

  If `-i` is specified, `-L` lists the policy table for a specific tunnel interface.

`-n`
: Show network addresses, ports, protocols in numbers. The `-n` option may only be used with the `-l` option.

`-q`
: Quiet mode. Suppresses the warning message generated when adding policies.

-r *file*
Remove IPsec policy rules from the system as specified by each entry in *file*. The format of the file contents is the same as is specified with the -a option. The file could be created with ipsecconf -l and then modified with an editor.

オペランド

Each policy entry contains three parts specified as follows:

```
{pattern} action {properties}
```

or

```
{pattern} action {properties} ["or" action {properties}]*
```

Every policy entry begins on a new line and can span multiple lines. If an entry exceeds the length of a line, you should split it only within a "braced" section or immediately before the first (left-hand) brace of a braced section. Avoid using the backslash character (\). See EXAMPLES.

The *pattern* section, as shown in the syntax above, specifies the traffic pattern that should be matched against the outbound and inbound datagrams. If there is a match, a specific *action* determined by the second argument will be taken, depending upon the *properties* of the policy entry.

If there is an or in the rule (multiple action-properties for a given pattern), a transmitter will use the first action-property pair that works, while a receiver will use any that are acceptable.

pattern and *properties* are name-value pairs where name and value are separated by a <space>, <tab> or <newline>. Multiple name-value pairs should be separated by <space>, <tab> or <newline>. The beginning and end of the pattern and properties are marked by { and } respectively.

Files can contain multiple policy entries. An unspecified name-value pair in the *pattern* will be considered as a wildcard. Wildcard entries match any corresponding entry in the datagram.

One thing to remember is that UDP port 500 is always bypassed regardless of any policy entries. This is a requirement for in.iked(1M) to work.

File can be commented by using a # as the first character. Comments may be inserted either at the beginning or the end of a line.

The complete syntax of a policy entry is:

```
policy ::= { <pattern1> } <action1> { <properties1> } |
     { <pattern2> } <action2> { <properties2> }
     [ 'or' <action2> { <properties2>} ]*

     pattern1 ::=  <pattern_name_value_pair1>*

     pattern2 ::=  <pattern_name_value_pair2>*
```

```
action1 ::= apply | permit | bypass | pass
action2 ::=  bypass | pass | drop | ipsec

properties1 ::=   {<prop_name_value_pair1>}
properties2 ::=   {<prop_name_value_pair2>}


pattern_name_value_pair1 ::=
   saddr <address>/<prefix> |
   src  <address>/<prefix> |
   srcaddr <address>/<prefix> |
   smask <mask> |
   sport <port> |
   daddr <address>/<prefix> |
   dst <address>/<prefix> |
   dstaddr <address>/<prefix> |
   dmask <mask> |
   dport <port> |
   ulp <protocol> |
   proto <protocol> |
   type <icmp-type> |
   type <number>-<number> |
   code <icmp-code>
   code <number>-<number>
   tunnel <interface-name> |
   negotiate <tunnel,transport>

pattern_name_value_pair2 ::=
   raddr <address>/<prefix> |
   remote <address>/<prefix> |
   rport <port> |
   laddr <address>/<prefix> |
   local <address>/<prefix> |
   lport <port> |
   ulp <protocol> |
   type <icmp-type> |
   type <number>-<number> |
   code <icmp-code> |
   code <number>-<number>
   proto <protocol>  |
   tunnel <interface-name> |
   negotiate <tunnel,transport> |
   dir <dir_val2>

address ::=  <IPv4 dot notation> | <IPv6 colon notation> |
             <String recognized by gethostbyname>|
             <String recognized by getnetbyname>
```

```
prefix ::=  <number>

mask ::= <0xhexdigit[hexdigit]> | <0Xhexdigit[hexdigit]> |
         <IPv4 dot notation>

port ::= <number>| <String recognized by getservbyname>

protocol ::=  <number>| <String recognized by getprotobyname>

prop_name_value_pair1 ::=
     auth_algs <auth_alg> |
     encr_algs <encr_alg> |
     encr_auth_algs <auth_alg> |
     sa <sa_val> |
     dir <dir_val1>

prop_name_value_pair2 ::=
     auth_algs <auth_alg> |
     encr_algs <encr_alg> |
     encr_auth_algs <auth_alg> |
     sa <sa_val>

auth_alg ::=  <auth_algname> ['(' <keylen> ')']
auth_algname ::= any | md5 | hmac-md5 | sha | sha1 | hmac-sha |
                 hmac-sha1 | hmac-sha256 | hmac-sha384 |
                 hmac-sha512 | aes-gmac128 | aes-gmac192 |
                 aes-gmac256 | <number>

encr_alg ::= <encr_algname> ['(' <keylen> ')']
encr_algname ::= any | aes | aes-cbc | des | des-cbc | 3des |
                 3des-cbc | blowfish | blowfish-cbc | aes-ccm |
                 aes-gcm | aes-none-gmac | <number>

keylen ::= <number> | <number>'..' | '..'<number> | <number>'..' \
<number>

sa_val ::= shared | unique

dir_val1 ::= out | in
dir_val2 ::= out | in | both

number ::= < 0 | 1 | 2 ... 9> <number>
icmp-type ::= <number> | unreach | echo | echorep | squench |
              redir | timex | paramprob | timest | timestrep |
              inforeq | inforep | maskreq | maskrep | unreach6 |
              pkttoobig6 | timex6 | paramprob6 | echo6 | echorep6 |
              router-sol6 | router-ad6 | neigh-sol6 | neigh-ad6 |
```

```
                redir6

icmp-code ::= <number> | net-unr | host-unr | proto-unr | port-unr |
              needfrag | srcfail | net-unk | host-unk | isolate |
              net-prohib | host-prohib | net-tos | host-tos |
              filter-prohib | host-preced | cutoff-preced |
              no-route6 | adm-prohib6 | addr-unr6 | port-unr6 |
              hop-limex6 | frag-re-timex6 | err-head6 | unrec-head6 |
              unreq-opt6
```

Policy entries may contain the following (name value) pairs in the *pattern* field. Each (name value) pair may appear only once in given policy entry.

laddr/plen
local/plen
: The value that follows is the local address of the datagram with the prefix length. Only plen leading bits of the source address of the packet will be matched. plen is optional. Local means destination on incoming and source on outgoing packets. The source address value can be a hostname as described in getaddrinfo(3SOCKET) or a network name as described in `getnetbyname(3XNET)` or a host address or network address in the Internet standard dot notation. See `inet_addr(3XNET)`. If a hostname is given and getaddrinfo(3SOCKET) returns multiple addresses for the host, then policy will be added for each of the addresses with other entries remaining the same.

raddr/plen
remote/plen
: The value that follows is the remote address of the datagram with the prefix length. Only plen leading bits of the remote address of the packet will be matched. plen is optional. Remote means source on incoming packets and destination on outgoing packets. The remote address value can be a hostname as described in `getaddrinfo(3SOCKET)` or a network name as described in `getnetbyname(3XNET)` or a host address or network address in the Internet standard dot notation. See `inet_addr(3XNET)`. If a hostname is given and `getaddrinfo(3SOCKET)` returns multiple addresses for the host, then policy will be added for each of the addresses with other entries remaining the same.

src/plen
srcaddr/plen
saddr/plen
: The value that follows is the source address of the datagram with the prefix length. Only *plen* leading bits of the source address of the packet will be matched. *plen* is optional.

  The source address value can be a hostname as described in `getaddrinfo(3SOCKET)` or a network name as described in `getnetbyname(3XNET)` or a host address or network address in the Internet standard dot notation. See `inet_addr(3XNET)`.

  If a hostname is given and `getaddrinfo(3SOCKET)` returns multiple addresses for the host, then policy will be added for each of the addresses with other entries remaining the same.

daddr/plen
dest/plen
dstaddr/plen
: The value that follows is the destination address of the datagram with the prefix length. Only *plen* leading bits of the destination address of the packet will be matched. *plen* is optional.

  See *saddr* for valid values that can be given. If multiple source and destination addresses are found, then a policy entry that covers each source address-destination address pair will be added to the system.

smask
: For IPv4 only. The value that follows is the source mask. If prefix length is given with *saddr*, this should not be given. This can be represented either in hexadecimal number with a leading `0x` or `0X`, for example, `0xffff0000`, `0Xffff0000` or in the Internet decimal dot notation, for example, `255.255.0.0` and `255.255.255.0`. The mask should be contiguous and the behavior is not defined for non-contiguous masks.

  smask is considered only when *saddr* is given.

  For both IPv4 and IPv6 addresses, the same information can be specified as a *slen* value attached to the *saddr* parameter.

dmask
: Analogous to *smask*.

lport
: The value that follows is the local port of the datagram. This can be either a port number or a string searched with a NULL proto argument, as described in getservbyname(3XNET)

rport
: The value that follows is the remote port of the datagram. This can be either a port number or a string searched with a NULL proto argument, as described in getservbyname(3XNET)

sport
: The value that follows is the source port of the datagram. This can be either a port number or a string searched with a `NULL` proto argument, as described in `getservbyname(3XNET)`

dport
: The value that follows is the destination port of the datagram. This can be either a port number or a string as described in `getservbyname(3XNET)` searched with `NULL` proto argument.

`proto` *ulp*
: The value that follows is the Upper Layer Protocol that this entry should be matched against. It could be a number or a string as described in `getprotobyname(3XNET)`. If no smask or plen is specified, a plen of 32 for IPv4 or 128 for IPv6 will be used, meaning a host. If the *ulp* is `icmp` or `ipv6-icmp`, any action applying IPsec must be the same for all `icmp` rules.

`type` *num* or *num-num*
: The value that follows is the ICMP type that this entry should be matched against. `type` must be a number from 0 to 255, or one of the appropriate `icmp-type` keywords. Also, *ulp* must be present and must specify either `icmp` or `ipv6-icmp`. A range of types can be specified with a hyphen separating numbers.

`code` *num* or *num-num*
: The value that follows is the ICMP code that this entry should be matched against. The value following the keyword `code` must be a number from 0 to 254 or one of the appropriate `icmp-code` keywords. Also, `type` must be present. A range of codes can be specified with a hyphen separating numbers.

`tunnel` *name*
: Specifies a tunnel network interface, as configured with `ifconfig(1M)`. If a tunnel of *name* does not yet exist, the policy entries are added anyway, and joined with the tunnel state when it is created. If a tunnel is unplumbed, its policy entries disappear.

`negotiate` *tunnel*
`negotiate` *transport*
: For per-tunnel security, specify whether the IPsec SAs protecting the traffic should be tunnel-mode SAs or transport-mode SAs. If transport-mode SAs are specified, no addresses can appear in the policy entry. Transport-mode is backward compatible with Solaris 9, and tunnel IPsec policies configured with `ifconfig(1M)` will show up as transport mode entries here.

Policy entries may contain the following (name-value) pairs in the properties field. Each (name-value) pair may appear only once in a given policy entry.

`auth_algs`
: An acceptable value following this implies that IPsec AH header will be present in the outbound datagram. Values following this describe the authentication algorithms that will be used while applying the IPsec AH on outbound datagrams and verified to be present on inbound datagrams. See RFC 2402.

  This entry can contain either a string or a decimal number.

  `string`
  : This should be either MD5 or HMAC-MD5 denoting the HMAC-MD5 algorithm as described in RFC 2403, and SHA1, or HMAC-SHA1 or SHA or HMAC-SHA denoting the HMAC-SHA algorithm described in RFC 2404. You can use the `ipsecalgs(1M)` command to obtain the complete list of authentication algorithms.

    The string can also be `ANY`, which denotes no-preference for the algorithm. Default algorithms will be chosen based upon the SAs available at this time for manual SAs and the key negotiating daemon for automatic SAs. Strings are not case-sensitive.

`number`
A number in the range 1-255. This is useful when new algorithms can be dynamically loaded.

If *auth_algs* is not present, the AH header will not be present in the outbound datagram, and the same will be verified for the inbound datagram.

`encr_algs`
An acceptable value following this implies that IPsec ESP header will be present in the outbound datagram. The value following this describes the encryption algorithms that will be used to apply the IPsec ESP protocol to outbound datagrams and verify it to be present on inbound datagrams. See RFC 2406.

This entry can contain either a string or a decimal number. Strings are not case-sensitive.

`string`
Can be one of the following:

string value:	Algorithm Used:	See RFC:
DES or DES-CBC	DES-CBC	2405
3DES or 3DES-CBC	3DES-CBC	2451
BLOWFISH or BLOWFISH-CBC	BLOWFISH-CBC	2451
AES or AES-CBC	AES-CBC	2451
AES-CCM	AES-CCM	4309
AES-GCM	AES_GCM	4106

You can use the `ipsecalgs(1M)` command to obtain the complete list of authentication algorithms.

The value can be `NULL`, which implies a `NULL` encryption, pursuant to RFC 2410. This means that the payload will not be encrypted. The string can also be `ANY`, which indicates no-preference for the algorithm. Default algorithms will be chosen depending upon the SAs available at the time for manual SAs and upon the key negotiating daemon for automatic SAs. Strings are not case-sensitive.

`number`
A decimal number in the range 1-255. This is useful when new algorithms can be dynamically loaded.

`encr_auth_algs`
An acceptable value following `encr_auth_algs` implies that the IPsec ESP header will be present in the outbound datagram. The values following `encr_auth_algs` describe the

authentication algorithms that will be used while applying the IPsec ESP protocol on outbound datagrams and verified to be present on inbound datagrams. See RFC 2406. This entry can contain either a string or a number. Strings are case-insensitive.

`string`
: Valid values are the same as the ones described for `auth_algs` above.

`number`
: This should be a decimal number in the range 1-255. This is useful when new algorithms can be dynamically loaded.

If `encr_algs` is present and `encr_auth_algs` is not present in a policy entry, the system will use an ESP SA regardless of whether the SA has an authentication algorithm or not.

If `encr_algs` is not present and `encr_auth_algs` is present in a policy entry, null encryption will be provided, which is equivalent to `encr_algs` with `NULL`, for outbound and inbound datagrams.

If both `encr_algs` and `encr_auth_algs` are not present in a policy entry, ESP header will not be present for outbound datagrams and the same will be verified for inbound datagrams.

If both `encr_algs` and `encr_auth_algs` are present in a policy entry, ESP header with integrity checksum will be present on outbound datagrams and the same will be verified for inbound datagrams.

For `encr_algs`, `encr_auth_algs`, and `auth_algs` a key length specification may be present. This is either a single value specifying the only valid key length for the algorithm or a range specifying the valid minimum and/or maximum key lengths. Minimum or maximum lengths may be omitted.

`dir`
: Values following this decides whether this entry is for outbound or inbound datagram. Valid values are strings that should be one of the following:

`out`
: This means that this policy entry should be considered only for outbound datagrams.

`in`
: This means that this policy entry should be considered only for inbound datagrams.

`both`
: This means that this policy entry should be considered for both inbound and outbound datagrams

This entry is not needed when the action is “apply”, “permit” or “ipsec”. But if it is given while the action is “apply” or “permit”, it should be “out” or “in” respectively. This is mandatory when the action is “bypass”.

`sa`

Values following this decide the attribute of the security association. Value indicates whether a unique security association should be used or any existing SA can be used. If there is a policy requirement, SAs are created dynamically on the first outbound datagram using the key management daemon. Static SAs can be created using `ipseckey(1M)`. The values used here determine whether a new SA will be used/obtained. Valid values are strings that could be one of the following:

`unique`

Unique Association. A new/unused association will be obtained/used for packets matching this policy entry. If an SA that was previously used by the same 5 tuples, that is, {Source address, Destination address, Source port, Destination Port, Protocol (for example, TCP/UDP)} exists, it will be reused. Thus uniqueness is expressed by the 5 tuples given above. The security association used by the above 5 tuples will not be used by any other socket. For inbound datagrams, uniqueness will not be verified.

For tunnel-mode tunnels, `unique` is ignored. SAs are assigned per-rule in tunnel-mode tunnels. For transport-mode tunnels, `unique` is implicit, because the enforcement happens only on the outer-packet addresses and protocol value of either IPv4-in-IP or IPv6-in-IP.

`shared`

Shared association. If an SA exists already for this source-destination pair, it will be used. Otherwise a new SA will be obtained. This is the default.

This is mandatory only for outbound policy entries and should not be given for entries whose action is “bypass”. If this entry is not given for inbound entries, for example, when “dir” is in or “action” is permit, it will be assumed to be shared.

Action follows the pattern and should be given before properties. It should be one of the following and this field is mandatory.

`ipsec`

Use IPsec for the datagram as described by the properties, if the pattern matches the datagram. If ipsec is given without a dir spec , the pattern is matched to incoming and outgoing datagrams.

`apply`

Apply IPsec to the datagram as described by the properties, if the pattern matches the datagram. If `apply` is given, the pattern is matched only on the outbound datagram.

`permit`

Permit the datagram if the pattern matches the incoming datagram and satisfies the constraints described by the properties. If it does not satisfy the properties, discard the datagram. If `permit` is given, the pattern is matched only for inbound datagrams.

`bypass`
`pass`

Bypass any policy checks if the pattern matches the datagram. `dir` in the properties decides whether the check is done on outbound or inbound datagrams. All the `bypass` entries are checked before checking with any other policy entry in the system. This has the highest precedence over any other entries. `dir` is the only field that should be present when action is `bypass`.

`drop`
Drop any packets that match the pattern.

If the file contains multiple policy entries, for example, they are assumed to be listed in the order in which they are to be applied. In cases of multiple entries matching the outbound and inbound datagram, the first match will be taken. The system will reorder the policy entry, that is, add the new entry before the old entry, only when:

The level of protection is "stronger" than the old level of protection.

Currently, strength is defined as:

```
AH and ESP > ESP > AH
```

The standard uses of AH and ESP were what drove this ranking of "stronger". There are flaws with this. ESP can be used either without authentication, which will allow cut-and-paste or replay attacks, or without encryption, which makes it equivalent or slightly weaker than AH. An administrator should take care to use ESP properly. See `ipsecesp(7P)` for more details.

If the new entry has `bypass` as action, `bypass` has the highest precedence. It can be added in any order, and the system will still match all the `bypass` entries before matching any other entries. This is useful for key management daemons which can use this feature to bypass IPsec as it protects its own traffic.

Entries with both AH (`auth_algs` present in the policy entry) and ESP (`encr_auth_algs` or `encr_auth_algs` present in the policy entry) protection are ordered after all the entries with AH and ESP and before any AH–only and ESP–only entries. In all other cases the order specified by the user is not modified, that is, newer entries are added at the end of all the old entries. See 使用例.

A new entry is considered duplicate of the old entry if an old entry matches the same traffic pattern as the new entry. See 使用例 for information on duplicates.

セキュリティー

If, for example, the policy file comes over the wire from an NFS mounted file system, an adversary can modify the data contained in the file, thus changing the policy configured on the machine to suit his needs. Administrators should be cautious about transmitting a copy of the policy file over a network.

To prevent non-privileged users from modifying the security policy, ensure that the configuration file is writable only by trusted users.

The configuration file is defined by a property of the `policy` smf(5) service. The default configuration file, is `/etc/inet/ipsecinit.conf`. This can be changed using the svcprop(1) command. See `NOTES` for more details.

The policy description language supports the use of tokens that can be resolved by means of a name service, using functions such as gethostbyname(3NSL). While convenient, these functions are only secure as the name service the system is configured to use. Great care should be taken to secure the name service if it is used to resolve elements of the security policy.

If your source address is a host that can be looked up over the network and your naming system itself is compromised, then any names used will no longer be trustworthy.

If the name switch is configured to use a name service that is not local to the system, bypass policy entries might be required to prevent the policy from preventing communication to the name service. See nsswitch.conf(4).

Policy is latched for TCP/UDP sockets on which a connect(3SOCKET) or accept(3SOCKET) has been issued. Adding new policy entries will not have any effect on them. This feature of latching may change in the future. It is not advisable to depend upon this feature.

The `ipsecconf` command can only be run by a user who has sufficient privilege to open the pf_key(7P) socket. The appropriate privilege can be assigned to a user with the Network IPsec Management profile. See profiles(1), rbac(5), prof_attr(4).

Make sure to set up the policies before starting any communications, as existing connections may be affected by the addition of new policy entries. Similarly, do not change policies in the middle of a communication.

Note that certain `ndd` tunables affect how policies configured with this tool are enforced; see ipsecesp(7P) for more details.

使用例

例 1 Protecting Outbound TCP Traffic With ESP and the AES Algorithm

The following example specified that any TCP packet from spiderweb to arachnid should be encrypted with AES, and the SA could be a shared one. It does not verify whether or not the inbound traffic is encrypted.

```
#
# Protect the outbound TCP traffic between hosts spiderweb
# and arachnid with ESP and use AES algorithm.
#
{
     laddr spiderweb
     raddr arachnid
     ulp tcp
     dir out
} ipsec {
```

例 1 Protecting Outbound TCP Traffic With ESP and the AES Algorithm (続き)

```
        encr_algs AES
}
```

例 2 Verifying Whether or Not Inbound Traffic is Encrypted

Example 1 does not verify whether or not the inbound traffic is encrypted. The entry in this example protects inbound traffic:

```
#
# Protect the TCP traffic on inbound with ESP/DES from arachnid
# to spiderweb
#
{
        laddr spiderweb
        raddr arachnid
        ulp tcp
        dir in
} ipsec {
        encr_algs AES
}
```

`sa` can be absent for inbound policy entries as it implies that it can be a shared one. Uniqueness is not verified on inbound. Note that in both the above entries, authentication was never specified. This can lead to cut and paste attacks. As mentioned previously, though the authentication is not specified, the system will still use an ESP SA with `encr_auth_alg` specified, if it was found in the SA tables.

例 3 Protecting All Traffic Between Two Hosts

The following example protects both directions at once:

```
{
        laddr spiderweb
        raddr arachnid
        ulp tcp
} ipsec {
        encr_algs AES
}
```

例 4 Authenticating All Inbound Traffic to the Telnet Port

This entry specifies that any inbound datagram to telnet port should come in authenticated with the SHA1 algorithm. Otherwise the datagram should not be permitted. Without this entry, traffic destined to port number 23 can come in clear. `sa` is not specified, which implies that it is shared. This can be done only for inbound entries. You need to have an equivalent entry to protect outbound traffic so that the outbound traffic is authenticated as well, remove the dir.

例 4 Authenticating All Inbound Traffic to the Telnet Port (続き)

```
#
# All the inbound traffic to the telnet port should be
# authenticated.
#
{
       lport telnet
       dir in
} ipsec {
       auth_algs sha1
}
```

例 5 Verifying Inbound Traffic is Null-Encrypted

The first entry specifies that any packet with address host-B should not be checked against any policies. The second entry specifies that all inbound traffic from network-B should be encrypted with a `NULL` encryption algorithm and the MD5 authentication algorithm. `NULL` encryption implies that ESP header will be used without encrypting the datagram. As the first entry is `bypass` it need not be given first in order, as `bypass` entries have the highest precedence. Thus any inbound traffic will be matched against all `bypass` entries before any other policy entries.

```
#
# Make sure that all inbound traffic from network-B is NULL
# encrypted, but bypass for host-B alone from that network.
# Add the bypass first.
{
raddr host-B
    dir in
} bypass {}

# Now add for network-B.
{
    raddr network-B/16
    dir in
} ipsec {
encr_algs NULL
encr_auth_algs md5
}
```

例 6 Entries to Bypass Traffic from IPsec

The first two entries provide that any datagram leaving the machine with source port 53 or coming into port number 53 should not be subjected to IPsec policy checks, irrespective of any other policy entry in the system. Thus the latter two entries will be considered only for ports other than port number 53.

例 6 Entries to Bypass Traffic from IPsec (続き)

```
#
# Bypass traffic for port no 53
     #
{lport 53} bypass {}
{rport 53} bypass {}
{raddr spiderweb } ipsec {encr_algs any sa unique}
```

例 7 Protecting Outbound Traffic

```
 #
     # Protect the outbound traffic from all interfaces.
     #
{raddr spiderweb dir out} ipsec {auth_algs any sa unique}
```

If the gethostbyname(3XNET) call for spiderweb yields multiple addresses, multiple policy entries will be added for all the source address with the same properties.

```
{
    laddr arachnid
    raddr spiderweb
    dir in
} ipsec {auth_algs any sa unique}
```

If the gethostbyname(3XNET) call for spiderweb and the gethostbyname(3XNET) call for arachnid yield multiple addresses, multiple policy entries will be added for each (saddr daddr) pair with the same properties. Use ipsecconf -l to view all the policy entries added.

例 8 Bypassing Unauthenticated Traffic

```
#
# Protect all the outbound traffic with ESP except any traffic
# to network-b which should be authenticated and bypass anything
# to network-c
#
{raddr network-b/16 dir out} ipsec {auth_algs any}
{dir out} ipsec {encr_algs any}
{raddr network-c/16 dir out} bypass {} # NULL properties
```

Note that bypass can be given anywhere and it will take precedence over all other entries. NULL pattern matches all the traffic.

例 9 Encrypting IPv6 Traffic with 3DES and MD5

The following entry on the host with the link local address fe80::a00:20ff:fe21:4483 specifies that any outbound traffic between the hosts wtih IPv6 link-local addresses fe80::a00:20ff:fe21:4483 and fe80::a00:20ff:felf:e346 must be encrypted with 3DES and MD5.

例 9 Encrypting IPv6 Traffic with 3DES and MD5 *(続き)*

```
{
    laddr fe80::a00:20ff:fe21:4483
    raddr fe80::a00:20ff:felf:e346
    dir out
} ipsec {
    encr_algs 3DES
    encr_auth_algs MD5
}
```

例 10 Verifying IPv6 Traffic is Authenticated with SHA1

The following two entries require that all IPv6 traffic to and from the IPv6 site-local network `fec0:abcd::0/32` be authenticated with SHA1.

```
{raddr fec0:abcd::0/32} ipsec { auth_algs SHA1 }
```

例 11 Key Lengths

```
# use aes at any key length
{raddr spiderweb} ipsec {encr_algs aes}

# use aes with a 192 bit key
{raddr spiderweb} ipsec {encr_algs aes(192)}

# use aes with any key length up to 192 bits
# i.e. 192 bits or less
{raddr spiderweb} ipsec {encr_algs aes(..192)}

# use aes with any key length of 192 or more
# i.e. 192 bits or more
{raddr spiderweb} ipsec {encr_algs aes(192..)}

#use aes with any key from 192 to 256 bits
{raddr spiderweb} ipsec {encr_algs aes(192..256)}

#use any algorithm with a key of 192 bits or longer
{raddr spiderweb} ipsec {encr_algs any(192..)}
```

例 12 Correct and Incorrect Policy Entries

The following are examples of correctly formed policy entries:

```
{ raddr that_system rport telnet } ipsec { encr_algs 3des encr_auth_algs
sha1 sa shared}

{
        raddr that_system
        rport telnet
```

例 12 Correct and Incorrect Policy Entries (続き)

```
} ipsec {
        encr_algs 3des
        encr_auth_algs sha1
        sa shared
}

{ raddr that_system rport telnet } ipsec
        { encr_algs 3des encr_auth_algs sha1 sa shared}

{ raddr that_system rport telnet } ipsec
        { encr_algs 3des encr_auth_algs sha1 sa shared} or ipsec
        { encr_algs aes encr_auth_algs sha1 sa shared}
```

...and the following is an incorrectly formed entry:

```
{ raddr that_system rport telnet } ipsec
        { encr_algs 3des encr_auth_algs sha1 sa shared}
        or ipsec { encr_algs aes encr_auth_algs sha1 sa shared}
```

In the preceding, incorrect entry, note that the third line begins with "`or ipsec`". Such an entry causes `ipsecconf` to return an error.

例 13 Allowing Neighbor Discovery to Occur in the Clear

The following two entries require that all IPv6 traffic to and from the IPv6 site-local network `fec0:abcd::0/32` be authenticated with SHA1. The second entry allows neighbor discovery to operate correctly.

```
{raddr fec0:abcd::0/32} ipsec { auth_algs SHA1 }
{raddr fec0:abcd::0/32 ulp ipv6-icmp type 133-137  dir both }
    pass { }
```

例 14 Using "or"

The following entry allows traffic using the AES or Blowfish algorithms from the remote machine spiderweb:

```
{raddr spiderweb} ipsec {encr_algs aes} or ipsec {encr_algs blowfish}
```

例 15 Configuring a Tunnel to be Backward-Compatible with Solaris 9

The following example is equivalent to "`encr_algs aes encr_auth_algs md5`" in `ifconfig(1M)`:

```
{tunnel ip.tun0 negotiate transport} ipsec {encr_algs aes
                                                encr_auth_algs md5}
```

例 16 Configuring a Tunnel to a VPN client with an Assigned Address

The following example assumes a distinct “inside” network with its own topology, such that a client's default route goes “inside”.

```
# Unlike route(1m), the default route has to be spelled-out.
{tunnel ip.tun0 negotiate tunnel raddr client-inside/32
laddr 0.0.0.0/0} ipsec {encr_algs aes encr_auth_algs sha1}
```

例 17 Transit VPN router between Two Tunnelled Subnets and a Third

The following example specifies a configuration for a VPN router that routes between two tunnelled subnets and a third subnet that is on-link. Consider remote-site A, remote-site B, and local site C, each with a `/24` address allocation.

```
# ip.tun0 between me (C) and remote-site A.
# Cover remote-site A to remote-side B.
{tunnel ip.tun0 negotiate tunnel raddr A-prefix/24 laddr
B-prefix/24} ipsec {encr_algs 3des encr_auth_algs md5}

# Cover remote-site A traffic to my subnet.
{tunnel ip.tun0 negotiate tunnel raddr A-prefix/24 laddr
C-prefix/24} ipsec {encr_algs 3des encr_auth_algs md5}

# ip.tun1 between me (C) and remote-site B.
# Cover remote-site B to remote-site A.
{tunnel ip.tun1 negotiate tunnel raddr B-prefix/24 laddr
A-prefix/24} ipsec {encr_algs aes encr_auth_algs sha1}

# Cover remote-site B traffic to my subnet.
{tunnel ip.tun1 negotiate tunnel raddr B-prefix/24 laddr
C-prefix/24} ipsec {encr_algs aes encr_auth_algs md5}
```

例 18 Using Combined Mode Ciphers

Combined mode ciphers provide data privacy and message authentication in a single operation. They are treated as special versions of `encr_algs`. They provide message authentication without the need to specify `encr_auth_algs`. The two combined mode ciphers supported are:

`aes-ccm`
: AES CCM Mode(Counter with CBC-MAC)

`aes-gcm`
: AES GCM Mode (Galois/Counter)

The parameters used are the same as any other encr_algs value. In both examples, the number in the algorithm token indicates the length of the Integrity Check Vector (ICV). See `ipsecalgs(1M)`.

例 18 Using Combined Mode Ciphers (続き)

```
# simple example using transport mode
{laddr 192.168.99.2 raddr 192.168.99.3} ipsec
    {encr_algs aes-gcm sa shared}
# simple example using CCM mode and 128 bit keys
{laddr 192.168.99.100 raddr 192.168.99.200} ipsec
    {encr_algs aes-ccm(128) sa shared}
```

例 19 Using AES GMAC

The AES GMAC algorithm is a hash algorithm that provides message authentication. An Integrity Check Vector (ICV) is calculated and transmitted as part of the authenticated packet.

The AES GMAC algorithm can be used in ESP mode, in which case the packet data and the ESP header are authenticated. When used with IPsec ESP, AES GMAC can only be specified as an encryption algorithm, even though it does not provide any encryption.

```
# simple example using AES GMAC and 128 bit keys for ESP
{laddr 192.168.99.100 raddr 192.168.99.200} ipsec
    {encr_algs aes-none-gmac(128) sa shared}
```

The above example is analogous to the following invalid example:

```
{laddr 192.168.99.100 raddr 192.168.99.200} ipsec
    {encr_algs null encr_auth_algs aes-gmac(128) sa shared}
```

When used with ESP, the `aes-none-gmac` algorithm takes the key length as an optional argument, the supported key lengths can be displayed using the `ipsecalgs(1M)` command.

The AES GMAC algorithm can be used in AH mode, in which case the whole packet, including the IP header, is authenticated. When used with IPsec AH, AES GMAC can only be specified as an authentication algorithm.

```
# simple example using AES GMAC and 128 bit keys for AH
{laddr 192.168.99.100 raddr 192.168.99.200} ipsec
    {auth_algs aes-gmac128 sa shared}
```

When used with AH, each key length has its own DOI number, the key length does not need to be specified as an argument.

ファイル

`/var/run/ipsecpolicy.conf`
: Cache of IPsec policies currently configured for the system, maintained by `ipsecconf` command. Do not edit this file.

`/etc/inet/ipsecinit.conf`
: File containing IPsec policies to be installed at system restart by the `policy` `smf(5)` service. See `NOTES` for more information.

`/etc/inet/ipsecinit.sample`
Sample input file for `ipseconf`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`auths(1)`, `profiles(1)`, `svcprop(1)`, `svcs(1)`, `in.iked(1M)`, `init(1M)`, `ifconfig(1M)`, `ipsecalgs(1M)`, `ipseckey(1M)`, `svcadm(1M)`, `svccfg(1M)`, `gethostbyname(3NSL)`, `accept(3SOCKET)`, `connect(3SOCKET)`, `gethostbyname(3XNET)`, `getnetbyname(3XNET)`, `getprotobyname(3XNET)`, `getservbyname(3XNET)`, `getaddrinfo(3SOCKET)`, `socket(3SOCKET)`, `ike.config(4)`, `nsswitch.conf(4)`, `prof_attr(4)`, `user_attr(4)`, `attributes(5)`, `rbac(5)`, `smf(5)`, `ipsecah(7P)`, `ipsecesp(7P)`, `pf_key(7P)`

Glenn, R. and Kent, S. RFC 2410, The NULL Encryption Algorithm and Its Use With IPsec. The Internet Society. 1998.

Kent, S. and Atkinson, R. RFC 2402, IP Authentication Header.The Internet Society. 1998.

Kent, S. and Atkinson, R. RFC 2406, IP Encapsulating Security Payload (ESP). The Internet Society. 1998.

Madsen, C. and Glenn, R. RFC 2403, The Use of HMAC-MD5-96 within ESP and AH. The Internet Society. 1998.

Madsen, C. and Glenn, R. RFC 2404, The Use of HMAC-SHA-1-96 within ESP and AH. The Internet Society. 1998.

Madsen, C. and Doraswamy, N. RFC 2405, The ESP DES-CBC Cipher Algorithm With Explicit IV. The Internet Society. 1998.

Pereira, R. and Adams, R. RFC 2451, The ESP CBC-Mode Cipher Algorithms. The Internet Society. 1998.

Frankel, S. and Kelly, R. Glenn, The AES Cipher Algorithm and Its Use With IPsec. 2001.

診断

Bad “string” on line *N*.
Duplicate “string” on line *N*.
string refers to one of the names in pattern or properties. A Bad string indicates that an argument is malformed; a Duplicate string indicates that there are multiple arguments of a similar type, for example, multiple Source Address arguments.

Interface name already selected
Dual use of `-i` *name* and *name*,*index* for an index.

Error before or at line *N*.
: Indicates parsing error before or at line *N*.

Non-existent index
: Reported when the *index* for delete is not a valid one.

spd_msg return: File exists
: Reported when there is already a policy entry that matches the traffic of this new entry.

注意事項

IPsec manual keys are managed by the service management facility, smf(5). The services listed below manage the components of IPsec. These services are delivered as follows:

```
svc:/network/ipsec/policy:default (enabled)
svc:/network/ipsec/ipsecalgs:default (enabled)
svc:/network/ipsec/manual-key:default (disabled)
svc:/network/ipsec/ike:default (disabled)
```

The manual-key service is delivered disabled. The system administrator must create manual IPsec Security Associations (SAs), as described in ipseckey(1M), before enabling that service.

The policy service is delivered enabled, but without a configuration file, so that, as a starting condition, packets are not protected by IPsec. After you create the configuration file `/etc/inet/ipsecinit.conf`, as described in this man page, and refresh the service (`svcadm refresh`, see below), the policy contained in the configuration file is applied. If there is an error in this file, the service enters maintenance mode.

Services that are delivered disabled are delivered that way because the system administrator must create configuration files for those services before enabling them. See ike.config(4) for the `ike` service.

See ipsecalgs(1M) for the `ipsecalgs` service.

The correct administrative procedure is to create the configuration file for each service, then enable each service using svcadm(1M).

If the configuration needs to be changed, edit the configuration file then refresh the service, as follows:

```
example# svcadm refresh policy
```

The smf(5) framework will record any errors in the service-specific log file. Use any of the following commands to examine the `logfile` property:

```
example# svcs -l policy
example# svcprop policy
example# svccfg -s policy listprop
```

The following property is defined for the `policy` service:

```
config/config_file
```

This property can be modified using svccfg(1M) by users who have been assigned the following authorization:

```
solaris.smf.value.ipsec
```

See auths(1), user_attr(4), rbac(5).

The service needs to be refreshed using svcadm(1M) before the new property is effective. General non-modifiable properties can be viewed with the svcprop(1) command.

```
# svccfg -s ipsec/policy setprop config/config_file = /new/config_file
# svcadm refresh policy
```

Administrative actions on this service, such as enabling, disabling, refreshing, and requesting restart can be performed using svcadm(1M). A user who has been assigned the authorization shown below can perform these actions:

```
solaris.smf.manage.ipsec
```

The service's status can be queried using the svcs(1) command.

The ipsecconf command is designed to be managed by the policy smf(5) service. While the ipsecconf command can be run from the command line, this is discouraged. If the ipsecconf command is to be run from the command line, the policy smf(5) service should be disabled first. See svcadm(1M).

名前

ipseckey – manually manipulate an IPsec Security Association Database (SADB)

形式

```
ipseckey [-nvp]
ipseckey [-nvp] -f filename
ipseckey -c filename
ipseckey [-nvp] [delete | delete-pair | get] SA_TYPE {EXTENSION value...}
ipseckey [-np] [monitor | passive_monitor | pmonitor]
ipseckey [-nvp] flush {SA_TYPE}
ipseckey [-nvp] dump {SA_TYPE}
ipseckey [-nvp] save SA_TYPE {filename}
ipseckey [-nvp] -s filename
```

機能説明

The `ipseckey` command is used to manually manipulate the security association databases of the network security services, `ipsecah(7P)` and `ipsecesp(7P)`. You can use the `ipseckey` command to set up security associations between communicating parties when automated key management is not available.

While the `ipseckey` utility has only a limited number of general options, it supports a rich command language. The user may specify requests to be delivered by means of a programmatic interface specific for manual keying. See `pf_key(7P)`. When `ipseckey` is invoked with no arguments, it will enter an interactive mode which prints a prompt to the standard output and accepts commands from the standard input until the end-of-file is reached. Some commands require an explicit security association (“SA”) type, while others permit the SA type to be unspecified and act on all SA types.

`ipseckey` uses a `PF_KEY` socket and the message types `SADB_ADD`, `SADB_DELETE`, `SADB_GET`, `SADB_UPDATE`, `SADB_FLUSH`, and `SADB_X_PROMISC`. Thus, you must be a superuser to use this command.

`ipseckey` handles sensitive cryptographic keying information. Please read the `Security` section for details on how to use this command securely.

オプション

`-c` [*filename*]
: Analogous to the `-f` option (see following), except that the input is not executed but only checked for syntactical correctness. Errors are reported to `stderr`. This option is provided to debug configurations without making changes. See `SECURITY` and “Service Management Facility” for more information.

`-f` [*filename*]
: Read commands from an input file, *filename*. The lines of the input file are identical to the command line language. The `load` command provides similar functionality. The `-s` option or the `save` command can generate files readable by the `-f` argument.

`-n`
Prevent attempts to print host and network names symbolically when reporting actions. This is useful, for example, when all name servers are down or are otherwise unreachable.

`-p`
Paranoid. Do not print any keying material, even if saving SAs. Instead of an actual hexadecimal digit, print an `X` when this flag is turned on.

`-s` [*filename*]
The opposite of the `-f` option. If '`-`' is given for a *filename*, then the output goes to the standard output. A snapshot of all current `SA` tables will be output in a form readable by the `-f` option. The output will be a series of `add` commands, but with some names not used. This occurs because a single name may often indicate multiple addresses.

`-v`
Verbose. Print the messages being sent into the `PF_KEY` socket, and print raw seconds values for lifetimes.

Commands

`add`
Add an SA. Because it involves the transfer of keying material, it cannot be invoked from the shell, lest the keys be visible in `ps(1)` output. It can be used either from the interactive `ipseckey>` prompt or in a command file specified by the `-f` command. The `add` command accepts all extension-value pairs described below.

`update`
Update SA lifetime, and in the cases of larval SAs (leftover from aborted automated key management), keying material and other extensions. Like `add`, this command cannot be invoked from the shell because keying material would be seen by the `ps(1)` command. It can be used either from the interactive `ipseckey>` prompt or in a command file specified by the `-f` command. The `update` command accepts all extension-value pairs, but normally is only used for SA lifetime updates.

`update-pair`
As update, but apply the update to the SA and its paired SA, if there is one.

`delete`
Delete a specific SA from a specific `SADB`. This command requires the `spi` extension, and the `dest` extension for IPsec SAs. Other extension-value pairs are superfluous for a delete message. If the SA to be deleted is paired with another SA, the SA is deleted and the paired SA is updated to indicate that it is now unpaired.

`delete-pair`
Delete a specific SA from a specific SADB. If the SA is paired with another SA, delete that SA too. This command requires the `spi` extension and the `dest` extension for the IPsec SA, or its pair.

`get`
Lookup and display a security association from a specific `SADB`. Like `delete`, this command only requires `spi` and `dest` for IPsec.

`flush`
: Remove all SA for a given `SA_TYPE`, or all SA for all types.

`monitor`
: Continuously report on any `PF_KEY` messages. This uses the `SADB_X_PROMISC` message to enable messages that a normal `PF_KEY` socket would not receive to be received. See `pf_key(7P)`.

`passive_monitor`
: Like monitor, except that it does not use the `SADB_X_PROMISC` message.

`pmonitor`
: Synonym for `passive_monitor`.

`dump`
: Will display all SAs for a given SA type, or will display all SAs. Because of the large amount of data generated by this command, there is no guarantee that all SA information will be successfully delivered, or that this command will even complete.

`save`
: Is the command analog of the `-s` option. It is included as a command to provide a way to snapshot a particular SA type, for example, `esp` or `ah`.

`help`
: Prints a brief summary of commands.

SA_TYPE

`all`
: Specifies all known SA types. This type is only used for the `flush` and `dump` commands. This is equivalent to having no SA type for these commands.

`ah`
: Specifies the IPsec Authentication Header (“`AH`”) SA.

`esp`
: Specifies the IPsec Encapsulating Security Payload (“`ESP`”) SA.

Extension Value Types

Commands like `add`, `delete`, `get`, and `update` require that certain extensions and associated values be specified. The extensions will be listed here, followed by the commands that use them, and the commands that require them. Requirements are currently documented based upon the IPsec definitions of an SA. Required extensions may change in the future. `<number>` can be in either hex (`0xnnn`), decimal (`nnn`) or octal (`0nnn`).`<string>` is a text string. `<hexstr>` is a long hexadecimal number with a bit-length. Extensions are usually paired with values; however, some extensions require two values after them.

`spi` *<number>*
: Specifies the security parameters index of the `SA`. This extension is required for the `add`, `delete`, `get` and `update` commands.

`pair-spi` <*number*>
: When `pair-spi` is used with the `add` or `update` commands, the SA being added or updated will be paired with the SA defined by `pair-spi`. A pair of SAs can be updated or deleted with a single command.

  The two SAs that make up the pair need to be in opposite directions from the same pair of IP addresses. The command will fail if either of the SAs specified are already paired with another SA.

  If the pair-spi token is used in a command and the SA defined by pair-spi does not exist, the command will fail. If the command was `add` and the pairing failed, the SA to be added will instead be removed.

`inbound | outbound`
: These optional flags specify the direction of the SA. When the `inbound` or `outbound` flag is specified with the `add` command, the kernel will insert the new SA into the specified hash table for faster lookups. If the flag is omitted, the kernel will decide into which hash table to insert the new SA based on its knowledge the IP addresses specified with the `src` and `dst` extensions.

  When these flags are used with the `update`, `delete`, `update-pair` or `get` commands, the flags provide a hint as to the hash table in which the kernel should find the SA.

`replay` <*number*>
: Specifies the replay window size. If not specified, the replay window size is assumed to be zero. It is not recommended that manually added SAs have a replay window. This extension is used by the `add` and `update` commands.

`replay_value` <*number*>
: Specifies the replay value of the SA. This extension is used by the `add` and `update` commands.

`state` <*string*>|<*number*>
: Specifies the SA state, either by numeric value or by the strings “`larval`”, “`mature`”, “`dying`” or “`dead`”. If not specified, the value defaults to `mature`. This extension is used by the `add` and `update` commands.

`auth_alg` <*string*>|<*number*>
`authalg <string>|<number>`
: Specifies the authentication algorithm for an `SA`, either by numeric value, or by strings indicating an algorithm name. Current authentication algorithms include:

  `HMAC-MD5`
  : `md5`, `hmac-md5`

  `HMAC-SH-1`
  : `sha`, `sha-1`, `hmac-sha1`, `hmac-sha`

HMAC-SHA-256
sha256, sha-256, hmac-sha256, hmac-sha-256

HMAC-SHA-384
sha384, sha-384, hmac-sha384, hmac-sha-384

HMAC-SHA-512
sha512, sha-512, hmac-sha512, hmac-sha-512

Often, algorithm names will have several synonyms. This extension is required by the add command for certain SA types. It is also used by the update command.

Use the ipsecalgs(1M) command to obtain the complete list of authentication algorithms.

encr_alg <*string*>|<*number*>
encralg <*string*>|<*number*>
Specifies the encryption algorithm for an SA, either by numeric value, or by strings indicating an algorithm name. Current encryption algorithms include DES ("des"), Triple-DES ("3des"), Blowfish ("blowfish"), and AES ("aes"). This extension is required by the add command for certain SA types. It is also used by the update command.

Use the ipsecalgs(1M) command to obtain the complete list of encryption algorithms.

The next six extensions are lifetime extensions. There are two varieties, "hard" and "soft". If a hard lifetime expires, the SA will be deleted automatically by the system. If a soft lifetime expires, an SADB_EXPIRE message will be transmitted by the system, and its state will be downgraded to dying from mature. See pf_key(7P). The monitor command to key allows you to view SADB_EXPIRE messages.

idle_addtime <*number*>
idle_usetime <*number*>
Specifies the number of seconds that this SA can exist if the SA is not used before the SA is revalidated. If this extension is not present, the default value is half of the hard_addtime (see below). This extension is used by the add and update commands.

soft_bytes <*number*>
hard_bytes <*number*>
Specifies the number of bytes that this SA can protect. If this extension is not present, the default value is zero, which means that the SA will not expire based on the number of bytes protected. This extension is used by the add and update commands.

soft_addtime <*number*>
hard_addtime <*number*>
Specifies the number of seconds that this SA can exist after being added or updated from a larval SA. An update of a mature SA does not reset the initial time that it was added. If this extension is not present, the default value is zero, which means the SA will not expire based on how long it has been since it was added. This extension is used by the add and update commands.

`soft_usetime` <*number*>
`hard_usetime` <*number*>
Specifies the number of seconds this SA can exist after first being used. If this extension is not present, the default value is zero, which means the SA will not expire based on how long it has been since it was added. This extension is used by the `add` and `update` commands.

`saddr` *address* | *name*
`srcaddr` *address* | *name*
`saddr6` *IPv6 address*
`srcaddr6` *IPv6 address*
`src` *address* | *name*
`src6` *IPv6 address*
`srcaddr` *address* and `src` *address* are synonyms that indicate the source address of the SA. If unspecified, the source address will either remain unset, or it will be set to a wildcard address if a destination address was supplied. To not specify the source address is valid for IPsec SAs. Future SA types may alter this assumption. This extension is used by the `add`, `update`, `get` and `delete` commands.

`daddr` <*address*>|<*name*>
`dstaddr` <*address*>|<*name*>
`daddr6` <*IPv6 address*>|<*name*>
`dstaddr6` <*IPv6 address*>|<*name*>
`dst` <*addr*>|<*name*>
`dst6` <*IPv6 address*>|<*name*>
`dstaddr` <*addr*> and `dst` <*addr*> are synonyms that indicate the destination address of the `SA`. If unspecified, the destination address will remain unset. Because IPsec SAs require a specified destination address and `spi` for identification, this extension, with a specific value, is required for the `add`, `update`, `get` and `delete` commands.

If a name is given, `ipseckey` will attempt to invoke the command on multiple SAs with all of the destination addresses that the name can identify. This is similar to how `ipsecconf` handles addresses.

If `dst6` or `dstaddr6` is specified, only the IPv6 addresses identified by a name are used.

`sport` <*portnum*>
`sport` specifies the source port number for an SA. It should be used in combination with an upper-layer protocol (see below), but it does not have to be.

`dport` <*portnum*>
sport specifies the destination port number for an SA. It should be used in combination with an upper-layer protocol (see below), but it does not have to be.

`encap` <*protocol*>
Identifies the protocol used to encapsulate NAT-traversal IPsec packets. Other NAT-traversal parameters (`nat_*`) are below. The only acceptable value for <*protocol*> currently is `udp`.

`proto` <*protocol number*>
`ulp` <*protocol number*>
`proto`, and its synonym `ulp`, specify the IP protocol number of the SA.

`nat_loc` <*address*>|<*name*>
If the local address in the SA (source or destination) is behind a NAT, this extension indicates the NAT node's globally-routable address. This address can match the SA's local address if there is a `nat_lport` (see below) specified.

`nat_rem` <*address*>|<*name*>
If the remote address in the SA (source or destination) is behind a NAT, this extension indicates that node's internal (that is, behind-the-NAT) address. This address can match the SA's local address if there is a `nat_rport` (see below) specified.

`nat_lport` <*portnum*>
Identifies the local UDP port on which encapsulation of ESP occurs.

`nat_rport` <*portnum*>
Identifies the remote UDP port on which encapsulation of ESP occurs.

`isrc` <*address*> | <*name*>[/<*prefix*>]
`innersrc` <*address*> | <*name*>[/<*prefix*>]
`isrc6` <*address*> | <*name*>[/<*prefix*>]
`innersrc6` <*address*> | <*name*>[/<*prefix*>]
`proxyaddr` <*address*> | <*name*>[/<*prefix*>]
`proxy` <*address*> | <*name*>[/<*prefix*>]
`isrc` <*address*>[/<*prefix*>] and `innersrc` <*address*>[/<*prefix*>] are synonyms. They indicate the inner source address for a tunnel-mode SA.

An inner-source can be a prefix instead of an address. As with other address extensions, there are IPv6-specific forms. In such cases, use only IPv6-specific addresses or prefixes.

Previous versions referred to this value as the proxy address. The usage, while deprecated, remains.

`idst` <*address*> | <*name*>[/<*prefix*>]
`innerdst` <*address*> | <*name*>[/<*prefix*>]
`idst6` <*address*> | <*name*>[/<*prefix*>]
`innerdst6` <*address*> | <*name*>[/<*prefix*>]
`idst` <*address*>[/<*prefix*>] and `innerdst` <*address*>[/<*prefix*>] are synonyms. They indicate the inner destination address for a tunnel-mode SA.

An inner-destination can be a prefix instead of an address. As with other address extensions, there are IPv6-specific forms. In such cases, use only IPv6-specific addresses or prefixes.

`innersport` <*portnum*>
`isport` <*portnum*>

`innersport` specifies the source port number of the inner header for a tunnel-mode SA. It should be used in combination with an upper-layer protocol (see below), but it does not have to be.

`innerdport` <*portnum*>
`idport` <*portnum*>
`innerdport` specifies the destination port number of the inner header for a tunnel-mode SA. It should be used in combination with an upper-layer protocol (see below), but it does not have to be.

`iproto` <*protocol number*>`iulp` <*protocol number*>
`iproto`, and its synonym `iulp`, specify the IP protocol number of the inner header of a tunnel-mode SA.

`authkey` <*hexstring*>
Specifies the authentication key for this SA. The key is expressed as a string of hexadecimal digits, with an optional `/` at the end, for example, `123/12`. Bits are counted from the most-significant bits down. For example, to express three '1' bits, the proper syntax is the string "`e/3`". For multi-key algorithms, the string is the concatenation of the multiple keys. This extension is used by the `add` and `update` commands.

`encrkey` <*hexstring*>
Specifies the encryption key for this SA. The syntax of the key is the same as `authkey`. A concrete example of a multi-key encryption algorithm is `3des`, which would express itself as a 192-bit key, which is three 64-bit parity-included `DES` keys. This extension is used by the `add` and `update` commands.

`reserved_bits` <*number*>
The last <*number*> bits of the `encrkey` string are marked as reserved in the `PF_KEY` message. This option is only for testing certain encryption algorithms.

Certificate identities are very useful in the context of automated key management, as they tie the `SA` to the public key certificates used in most automated key management protocols. They are less useful for manually added SAs. Unlike other extensions, `srcidtype` takes two values, a *type*, and an actual *value*. The type can be one of the following:

`prefix`
An address prefix.

`fqdn`
A fully-qualified domain name.

`domain`
Domain name, synonym for `fqdn`.

`user_fqdn`
User identity of the form *user@fqdn*.

`mailbox`
Synonym for `user_fqdn`.

The *value* is an arbitrary text string that should identify the certificate.

`srcidtype` <*type, value*>
: Specifies a source certificate identity for this SA. This extension is used by the `add` and `update` commands.

`dstidtype` <*type, value*>
: Specifies a destination certificate identity for this SA. This extension is used by the `add` and `update` commands

Label extensions are used on Trusted Extensions to associate sensitivity labels with the traffic carried inside a security association. These extensions are not allowed unless Trusted Extensions is enabled.

`label` *label*
: Defines the sensitivity label of traffic carried by this SA. Disallowed on systems not using Trusted Extensions.

`outer-label` *label*
: Defines the sensitivity of the ciphertext traffic belonging to this SA; this label will appear in the outer packet header. Disallowed on systems not using Trusted Extensions. Incorrect use of this extension might allow label policy to be circumvented.

`implicit-label` *label*
: Defines the sensitivity of the ciphertext traffic belonging to this SA and request that this SA not contain an explicit on-the-wire label. Disallowed on systems not using Trusted Extensions. Incorrect use of this extension might allow label policy to be circumvented.

Tunnel Mode versus Transport Mode SAs

An IPsec SA is a Tunnel Mode SA if the "proto" value is either 4 (`ipip`) or 41 (`ipv6`) *and* there is an inner-address or inner-port value specified. Otherwise, the SA is a Transport Mode SA.

セキュリティー

Keying material is very sensitive and should be generated as randomly as possible. Some algorithms have known weak keys. IPsec algorithms have built-in weak key checks, so that if a weak key is in a newly added `SA`, the `add` command will fail with an invalid value.

The `ipseckey` command allows a privileged user to enter cryptographic keying information. If an adversary gains access to such information, the security of IPsec traffic is compromised. The following issues should be taken into account when using the `ipseckey` command.

1. Is the TTY going over a network (interactive mode)?
   - If it is, then the security of the keying material is the security of the network path for this TTY's traffic. Using `ipseckey` over a clear-text `telnet` or `rlogin` session is risky.
   - Even local windows might be vulnerable to attacks where a concealed program that reads window events is present.
2. Is the file accessed over the network or readable to the world (`-f` option)?
   - A network-mounted file can be sniffed by an adversary as it is being read.
   - A world-readable file with keying material in it is also risky.

3. The `ipseckey` command is designed to be managed by the `manual-key` `smf(5)` service. Because the `smf(5)` log files are world-readable, the `ipseckey` does not record any syntax errors in the log files, as these errors might include secret information.

   If a syntax error is found when the `manual-key` `smf(5)` service is enabled, the service enters maintenance mode. The log file will indicate that there was a syntax error, but will not specify what the error was.

   The administrator should use `ipeckey -c` *filename* from the command line to discover the cause of the errors. See `OPTIONS`.

If your source address is a host that can be looked up over the network and your naming system itself is compromised, then any names used will not be trustworthy.

Security weaknesses often lie in misapplication of tools, not in the tools themselves. Administrators are urged to be cautious when using `ipseckey`. The safest mode of operation is probably on a console or other hard-connected TTY.

For further thoughts on this subject, see the afterward by Matt Blaze in Bruce Schneier's Applied Cryptography: Protocols, Algorithms, and Source Code in C.

Service Management Facility

IPsec manual keys are managed by the service management facility, `smf(5)`. The services listed below manage the components of IPsec. These services are delivered as follows:

```
svc:/network/ipsec/policy:default (enabled)
svc:/network/ipsec/ipsecalgs:default (enabled)
svc:/network/ipsec/manual-key:default (disabled)
svc:/network/ipsec/ike:default (disabled)
```

The manual-key service is delivered disabled. The system administrator must create manual IPsec Security Associations (SAs), as described in this man page, before enabling that service.

The policy service is delivered enabled, but without a configuration file, so that, as a starting condition, packets are not protected by IPsec. After you create the configuration file `/etc/inet/ipsecinit.conf` and refresh the service (`svcadm refresh`, see below), the policy contained in the configuration file is applied. If there is an error in this file, the service enters maintenance mode. See `ipsecconf(1M)`.

Services that are delivered disabled are delivered that way because the system administrator must create configuration files for those services before enabling them. See `ike.config(4)` for the `ike` service.

See `ipsecalgs(1M)` for the `ipsecalgs` service.

The correct administrative procedure is to create the configuration file for each service, then enable each service using `svcadm(1M)`.

If the configuration needs to be changed, edit the configuration file then refresh the service, as follows:

```
example# svcadm refresh manual-key
```

Warning: To prevent ipseckey complaining about duplicate Associations, the ipseckey command flushes the Security Association Data Base (SADB) when the ipseckey command is run from smf(5), before adding any new Security Associations defined in the configuration file. This differs from the command line behavior where the SADB is not flushed before adding new Security Associations.

The smf(5) framework will record any errors in the service-specific log file. Use any of the following commands to examine the logfile property:

```
example# svcs -l manual-key
example# svcprop manual-key
example# svccfg -s manual-key listprop
```

The following property is defined for the manual-key service:

```
config/config_file
```

This property can be modified using svccfg(1M) by users who have been assigned the following authorization:

```
solaris.smf.value.ipsec
```

See auths(1), user_attr(4), rbac(5).

The service needs to be refreshed using svcadm(1M) before the new property is effective. General non-modifiable properties can be viewed with the svcprop(1) command.

```
# svccfg -s ipsec/manual-key setprop config/config_file = \
/new/config_file
# svcadm refresh manual-key
```

Administrative actions on this service, such as enabling, disabling, refreshing, and requesting restart can be performed using svcadm(1M). A user who has been assigned the authorization shown below can perform these actions:

```
solaris.smf.manage.ipsec
```

The service's status can be queried using the svcs(1) command.

The ipseckey command is designed to be run under smf(5) management. While the ipsecconf command can be run from the command line, this is discouraged. If the ipseckey command is to be run from the command line, the manual-key smf(5) service should be disabled first. See svcadm(1M).

使用例

例 1　Emptying Out All SAs

To empty out all SA:

```
example# ipseckey flush
```

例 2 Flushing Out IPsec AH SAs Only

To flush out only IPsec AH SAs:

```
example# ipseckey flush ah
```

例 3 Saving All SAs To Standard Output

To save all SAs to the standard output:

```
example# ipseckey save all
```

例 4 Saving ESP SAs To The File /tmp/snapshot

To save ESP SAs to the file /tmp/snapshot:

```
example# ipseckey save esp /tmp/snapshot
```

例 5 Deleting an IPsec SA

To delete an IPsec SA, only the SPI and the destination address are needed:

```
example# ipseckey delete esp spi 0x2112 dst 224.0.0.1
```

An alternative would be to delete the SA and the SAs pair if it has one:

```
example# ipseckey delete-pair esp spi 0x2112 dst 224.0.0.1
```

例 6 Getting Information on an IPsec SA

Likewise, getting information on a SA only requires the destination address and SPI:

```
example# ipseckey get ah spi 0x5150 dst mypeer
```

例 7 Adding or Updating IPsec SAs

Adding or updating SAs requires entering interactive mode:

```
example# ipseckey
ipseckey> add ah spi 0x90125 src me.domain.com dst you.domain.com \
          authalg md5 authkey 1234567890abcdef1234567890abcdef
ipseckey> update ah spi 0x90125 dst you.domain.com hard_bytes \
          16000000
ipseckey> exit
```

Adding two SAs that are linked together as a pair:

```
example# ipseckey
ipseckey> add esp spi 0x2345 src me.domain.com dst you.domain.com \
   authalg md5 authkey bde359723576fdea08e56cbe876e24ad \
   encralg des encrkey be02938e7def2839
ipseckey> add esp spi 0x5432 src me.domain.com dst you.domain.com \
   authalg md5 authkey bde359723576fdea08e56cbe876e24ad \
   encralg des encrkey be02938e7def2839 pair-spi 0x2345
```

例 7 Adding or Updating IPsec SAs *(続き)*

```
ipseckey> exit
```

例 8 Adding an SA in the Opposite Direction

In the case of IPsec, SAs are unidirectional. To communicate securely, a second SA needs to be added in the opposite direction. The peer machine also needs to add both SAs.

```
example# ipseckey
ipseckey> add ah spi 0x2112 src you.domain.com dst me.domain.com \
          authalg md5 authkey bde359723576fdea08e56cbe876e24ad \
          hard_bytes 16000000
ipseckey> exit
```

例 9 Monitoring PF_KEY Messages

Monitoring for PF_KEY messages is straightforward:

```
example# ipseckey monitor
```

例 10 Using Commands in a File

Commands can be placed in a file that can be parsed with the -f option. This file may contain comment lines that begin with the "#" symbol. For example:

```
# This is a sample file for flushing out the ESP table and
# adding a pair of SAs.

flush esp

### Watch out!  I have keying material in this file.  See the
### SECURITY section in this manual page for why this can be
### dangerous .

add esp spi 0x2112 src me.domain.com dst you.domain.com \
    authalg md5 authkey bde359723576fdea08e56cbe876e24ad \
    encralg des encrkey be02938e7def2839 hard_usetime 28800
add esp spi 0x5150 src you.domain.com dst me.domain.com \
    authalg md5 authkey 930987dbe09743ade09d92b4097d9e93 \
    encralg des encrkey 8bd4a52e10127deb hard_usetime 28800

## End of file  -  This is a gratuitous comment
```

例 11 Adding SAs for IPv6 Addresses

The following commands from the interactive-mode create an SA to protect IPv6 traffic between the site-local addresses

```
example # ipseckey
ipseckey> add esp spi 0x6789 src6 fec0:bbbb::4483 dst6 fec0:bbbb::7843\
           authalg md5 authkey bde359723576fdea08e56cbe876e24ad \
```

例 11 Adding SAs for IPv6 Addresses (続き)

```
          encralg des encrkey be02938e7def2839 hard_usetime 28800
ipseckey>exit
```

例 12 Linking Two SAs as a Pair

The following command links two SAs together, as a pair:

```
example# ipseckey update esp spi 0x123456 dst 192.168.99.2 \
pair-spi 0x654321
```

ファイル

`/etc/inet/secret/ipseckeys`
: Default configuration file used at boot time. See "Service Management Facility" and `SECURITY` for more information.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`ps(1)`, `svcprop(1)`, `svcs(1)`, `ipsecconf(1M)`, `ipsecalgs(1M)`, `route(1M)`, `svcadm(1M)`, `svccfg(1M)`, `ike.config(4)`, `attributes(5)`, `smf(5)`, `ipsec(7P)`, `ipsecah(7P)`, `ipsecesp(7P)`, `pf_key(7P)`

Schneier, B., Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second ed. New York, New York: John Wiley & Sons, 1996.

診断

The `ipseckey` command parses the configuration file and reports any errors. In the case of multiple errors, `ipseckey` reports as many of these as possible.

The `ipseckey` command does not attempt to use a `COMMAND` that has a syntax error. A `COMMAND` might be syntactically correct but can nevertheless generate an error because the kernel rejected the request made to `pf_key(7P)`. This might occur because a key had an invalid length or because an unsupported algorithm was specified.

If there are any errors in the configuration file, ipseckey reports the number of valid COMMANDS and the total number of COMMANDS parsed.

`Parse error on line` *N*.
: If an interactive use of `ipseckey` would print usage information, this would print instead. Usually proceeded by another diagnostic. Because `COMMANDS` can cover more than a single line in the configuration file by using the backslash character to delimit lines, its not always possible to pinpoint in the configuration file the exact line that caused the error.

`Unexpected end of command line.`
: An additional argument was expected on the command line.

Unknown
: A value for a specific extension was unknown.

`Address type` *N* `not supported.`
: A name-to-address lookup returned an unsupported address family.

N `is not a bit specifier`
`bit length` *N* `is too big for`
`string is not a hex string`
: Keying material was not entered appropriately.

`Can only specify single`
: A duplicate extension was entered.

`Don't use extension for` <*string*> `for` <*command*>`.`
: An extension not used by a command was used.

`One of the entered values is incorrect: Diagnostic code` *NN*`:` <*msg*>
: This is a general invalid parameter error. The diagnostic code and message provides more detail about what precise value was incorrect and why.

注意事項

In spite of its IPsec-specific name, `ipseckey` is analogous to `route(1M)`, in that it is a command-line interface to a socket-based administration engine, in this case, `PF_KEY`. `PF_KEY` was originally developed at the United States Naval Research Laboratory.

To have machines communicate securely with manual keying, SAs need to be added by all communicating parties. If two nodes wish to communicate securely, both nodes need the appropriate SAs added.

In the future `ipseckey` may be invoked under additional names as other security protocols become available to `PF_KEY`.

This command requires `sys_ip_config` privilege to operate and thus can run in the global zone and in exclusive-IP zones. The global zone can set up security associations with `ipseckey` to protect traffic for shared-IP zones on the system.

名前

iscsiadm – iSCSI イニシエータの管理の有効化

形式

`iscsiadm` *subcommand direct-object* [*options*] [*operand*]

機能説明

`iscsiadm` コマンドは、ホスト上で iSCSI (インターネット SCSI) イニシエータを管理できるようにします。`iscsiadm` は、独自のオプションを備えた一連のサブコマンドとして実装されています。これらのオプションについては、各サブコマンドのセクションで説明します。特定のサブコマンドに関連付けられていないオプションは、「オプション」で説明されています。

`iscsiadm` は、次のサービスがオンラインの場合にのみ動作します。

`svc:/network/iscsi/initiator:default`

`iscsiadm` コマンドは次のサブコマンドをサポートしています。各サブコマンドについては、以降のサブセクションで詳細に説明します。

`add` オブジェクトに要素を追加します。

`list` オブジェクトの要素を一覧表示します。

`modify` オブジェクトの属性を変更します。

`remove` オブジェクトから要素を削除します。

`iscsiadm` サブコマンドは、直接オブジェクト (*direct-object*) に対して動作します。これらについては、各サブコマンドのセクションで説明します。

`iscsiadm` コマンドでは、iSCSI ターゲットの発見のために iSNS (Internet Storage Name Service) がサポートされています。このコマンドではまた、認証のために CHAP (Challenge Handshake Authentication Protocol) がサポートされています。

add サブコマンド

`add` サブコマンドの構文は次のとおりです。

`# iscsiadm add` *direct_object* [*operands...*]

`add` サブコマンドは、次の直接オブジェクトを追加します。

`discovery-address` *discovery-address* [...]

発見アドレスのリストにターゲットを追加します。発見アドレス (下に示す構文で指定される) は、SendTargets 発見セッションで使用される IP *address*:*port* の組み合わせです。この発見アプローチを使用すると、ターゲットデバイスはターゲットアドレスと、そのデバイスによって公開された各ターゲットのターゲット名をイニシエータに通知できます。ターゲットへの接続は、SendTargets 発見方式がホスト上で有効になっていないかぎり試行されません。この方式は `modify` サブコマンドで有効にします。

discovery-address パラメータの形式は次のとおりです。

<*IP address*>[:*port*]

port が指定されていない場合は、3260 のデフォルト値が使用されます。

`isns-server` *isns-server* [...]
iSNS サーバーアドレスのリストに iSNS サーバーを追加します。iSNS サーバーアドレス (下に示す構文で指定される) は、iSNS 発見セッションで使用される IP アドレスとポートの組み合わせです。iSNS 発見を使用すると、iSNS サーバーはイニシエータにポータルに関する情報や、そのイニシエータと同じ発見ドメインに属する各ターゲットの名前を提供できます。iSNS サーバーへの接続は、iSNS 発見方式がホスト上で有効になっていないかぎり、試行されません。この方式は、後述の `modify` サブコマンドで有効にします。

isns-server パラメータの形式は次のとおりです。

IP_address`[:`*port*`]`

ポートが指定されていない場合は、3205 のデフォルト値が使用されます。

`static-config` *static_target* [...]
静的に設定されたターゲットのリストにターゲットを追加します。ターゲットへの接続は、静的な設定による発見方式が有効になっていないかぎり試行されません。

static_target パラメータの形式は次のとおりです。

<*target-name*>`,`<*target address*>`[:`*port-number*`][,`*tpgt*`]`

<*target-name*> には最大 223 文字を指定できます。

list サブコマンド

`list` サブコマンドの構文は次のとおりです。

```
# iscsiadm list direct-object [options]
```

`list` サブコマンドは、次の直接オブジェクトのデータを表示します。

`discovery`
発見方式と、それらの現在の有効化状態 (有効または無効) を一覧表示します。発見方式には次のものがあります。

- iSNS (Internet Storage Name Service)
- 静的
- SendTargets

`initiator-node`
ホスト上のイニシエータノードに関する情報を一覧表示します。iSCSI イニシエータノードは論理 HBA を表し、iSCSI ターゲットのための論理的なホスト接続ポイントです。応答に一覧表示されるパラメータ値は、このイニシエータのデフォルトのパラメータ設定です。イニシエータに接続されている各ターゲットには、イニシエータノード上のパラメータ値とは異なるパラメータ値を設定できます。

`static-config` [*static_target*[, ...]]
指定されたターゲットのターゲット名とアドレスを一覧表示します。静的なターゲットが指定されていない場合は、静的に検出されたすべてのターゲットが表示されます。

`target` [`-S`] [`-v`] [*target*[, ...]]
ターゲットの現在のパラメータ、接続状態、およびそのターゲットの発見に使用された方式を一覧表示します。指定されたターゲットに関する情報を報告します。ターゲットが指定されていない場合は、検出されたか、または `modify target` サブコマンドによって変更されたパラメータを持っていたすべてのターゲットが報告されます。

指定されたターゲットに対して `-S` オプションとともに使用された場合は、次の情報が返されます。

- ターゲット名
- 論理ユニット番号
- ベンダー ID
- 製品 ID
- OS デバイス名 (たとえば、`/dev/rdsk/c0t2d0s0`)

`-v` オプションを指定すると、現在のログインパラメータ、詳細な接続情報、そのターゲットの発見に使用された発見方式などの、さらに詳細な情報が返されます。

発見方式のパラメータとして `NA` が返された場合は、そのターゲットが `iscsiadm modify target-param` コマンドを使用して作成され、検出されたオブジェクトとしては存在していないことを示します。このようなターゲットを削除するには、`iscsiadm remove target-param` を使用します。

`target-param` [`-v`] *target* [...]
ターゲットのデフォルトパラメータおよびユーザー定義パラメータを一覧表示します。

`discovery-address` [`-v`] [*discovery-address*[, ...]]
`iscsiadm add discovery-address` サブコマンドを使用して追加された `discovery-address` オブジェクトを一覧表示します。

`-v` オプションとともに使用された場合は、指定された *discovery-address* にあるすべての既知のターゲットを一覧表示します。`-v` オプションを指定すると、1 つ以上のターゲット名が、0 個以上のターゲットアドレス、および関連付けられたターゲットポータルグループタグ (TPGT) (該当する場合) とともに返されます。

`isns-server` [`-v`] [*isns-server*[, ...]]
`iscsiadm add isns-server` サブコマンドを使用して追加された `isns-server` オブジェクトを一覧表示します。

-v オプションとともに使用された場合は、指定された isns-server アドレスにあるすべての既知のターゲットを一覧表示します。 -v オプションを指定すると、1 つ以上のターゲット名が、0 個以上のターゲットアドレス、および関連付けられたターゲットポータルグループタグ (該当する場合) とともに返されます。

modify サブコマンド

modify サブコマンドの構文は次のとおりです。

```
# iscsiadm modify direct_object [options]
```

modify サブコマンドは、次の直接オブジェクトをサポートしています。

discovery [*options*]
: 発見方式を有効にすると、その方式を使用した発見が開始されます。現在有効になっている発見方式を無効にしても、その方式によってすでに検出されたターゲットへの接続には影響を与えません。

  modify discovery のオプションは次のとおりです。

  -i, -iSNS enable | disable
  : iSNS 発見を有効または無効にします。

  -s, --static enable | disable
  : 静的な発見を有効または無効にします。

  -t, --sendtargets enable | disable
  : SendTargets 発見を有効または無効にします。

initiator-node [*options*]
: イニシエータのプロパティーを変更します。ターゲットが現在接続されている場合、この操作は成功します。ただし、変更された一連のパラメータは、既存の接続セッションが存在しなくなって新しい接続が確立されるまで、そのターゲットに対して有効になりません。オプション -C と --CHAP-secret を指定した場合は、プロンプトで CHAP シークレットを入力する必要があります。

  Solaris I/O マルチパス機能 (以前は Sun StorEdge Traffic Manager [STMS] または MPxIO と呼ばれていた) が無効になっている場合の iSCSI ブートでは、次の initiator-node オプションだけを変更できます。

  - -r, --radius-server
  - -R, --radius-access
  - -P, --radius-shared-secret

  Solaris I/O マルチパス機能が有効になっている場合の iSCSI ブートでは、次の initiator-node オプションだけを変更できます。

  - -h, --headerdigest
  - -d, --datadigest
  - -c, --configured-sessions

  modify initiator-node のオプションは次のとおりです。

-A, --node-alias <*initiator node alias*>
イニシエータノードの別名を変更します。最大長は 223 文字です。

-a, --authentication chap | none
認証モードを設定します。

-C, --CHAP-secret
CHAP シークレットの値を設定します。デフォルト値はありません。最大長は 16 文字、最低限必要な長さは 12 文字です。

-c, --configured-sessions <*num_sessions*> | <*IP address*>[,<*IP address*>...]
各 iSCSI ターゲットに対して作成される設定済みの iSCSI セッションの数を設定します。この機能は、scsi_vhci(7D) で説明されている Solaris I/O マルチパス機能と組み合わせて使用してください。

-d, --datadigest none | CRC32
SCSI データ転送をチェックするために CRC32 を有効にするかどうかを設定します。

-H, --CHAP-name *CHAP name*
CHAP ユーザー名を指定します。このオプションを使用しない場合は、初期化時に CHAP 名がイニシエータノード名に設定されます。認証方式が CHAP に設定されている場合 (上の -a/--authentication オプションを参照)、CHAP ユーザー名は iscsiadm list initiator-node コマンドで表示されます。

-h, --headerdigest none | CRC32
SCSI パケットヘッダーをチェックするために CRC32 を有効にするかどうかを設定します。

-m, --max-connections *number_connections*
iSCSI セッションの最大接続数を変更します。デフォルト値は 1 です。各セッションの最大接続数は 65535 です。

-N, --node-name <*initiator node name*>
イニシエータノード名を変更します。最大長は 223 文字です。

注 – Solaris のインストール中に、イニシエータノード名はグローバルに一意な値に設定されます。この値を変更すると、iSCSI ネットワーク内の操作に悪影響を与えることがあります。

-P, --radius-shared-secret (exclusive)
RADIUS 共有シークレットを設定します。

-R, --radius-access enable | disable
RADIUS サーバーを使用するかどうかを設定します。

-r, --radius-server <*IP address*>[:<*port*>]
使用する RADIUS サーバーの IP アドレスとポートを設定します。

-T, --tunable-param <<*tunable-prop*>=<*value*>, ...>
イニシエータノードが接続されているすべてのターゲットに対して 1 つ以上の調整可能なパラメータを指定します。

注 – これらの値は、iSCSI ネットワーク内におけるこのパラメータの影響についての実際的な知識を備えた管理者だけが変更するようにしてください。

サポートされている tunable-prop オプションは次のとおりです。

recv-login-rsp-timeout
セッションログインの応答時間

recv-login-rsp-timeout オプションは、iSCSI イニシエータが iSCSI ターゲットからの iSCSI セッションログイン要求の応答を待つ時間を指定します。有効な値は 0 〜 60*60 であり、デフォルト値は 60 秒です。

conn-login-max
接続再試行の最大時間

conn-login-max オプションを指定すると、指定された時間ウィンドウ中に入出力タイムアウトまたは接続の失敗が発生した場合に、iSCSI イニシエータはターゲットへの接続を再確立できます。有効な値は 0 〜 60*60 であり、デフォルト値は 180 秒です。

polling-login-delay
ログイン再試行の時間間隔

polling-login-delay オプションは、入出力タイムアウトまたは接続の失敗が発生した場合の iSCSI イニシエータからターゲットへの各ログイン再試行の時間間隔を指定します。有効な値は 0 〜 60*60 であり、デフォルト値は 60 秒です。

target-param [*options*] *target*
ターゲットのパラメータを変更します。ターゲットが現在接続されている場合、この変更操作は成功します。ただし、変更された設定が有効になるまでに数秒かかることがあります。これらの設定が有効なことを確認するには、iscsiadm list target -v を使用します。指定されたターゲットがどの発見方式にも関連付けられていない場合は、指定されたパラメータを使用してターゲットオブジェクトが作成されます。このコマンドを使用してターゲットのパラメータを変更したあと、新しいパラメータは、そのターゲット上で iscsiadm remove target-param コマンドで変更または削除されるまで持続します。オプション -C と --CHAP-secret を指定した場合は、プロンプトで CHAP シークレットを入力する必要があります。

modify target-param のオプションは次のとおりです。

-B, --bi-directional-authentication enable | disable
双方向オプションを設定します。enable に設定されている場合、イニシエータは、指定されたターゲットに対して双方向認証を実行します。

`-C, --CHAP-secret`
ターゲットの CHAP シークレットの値を設定します。デフォルト値はありません。指定できる最大長は 16 文字です。

`-c, --configured-sessions` <*num_sessions*> | <*IP address*>[,<*IP address*>...]
各 iSCSI ターゲットに対して作成される設定済みの iSCSI セッションの数を設定します。この機能は、`scsi_vhci(7D)` で説明されている Solaris I/O マルチパス機能と組み合わせて使用してください。

`-d, --datadigest none | CRC32`
データのために CRC32 を有効または無効のどちらにするかを設定します。

`-H, --CHAP-name` *CHAP name*
CHAP ユーザー名を設定します。このオプションを使用しない場合は、初期化時に CHAP 名がターゲット名に設定されます。認証方式が CHAP に設定されている場合 (上の `initiator-node` 直接オブジェクトの下にある `-a/--authentication` オプションを参照)、CHAP ユーザー名は `iscsiadm list initiator-node` コマンドで表示されます。

`-h, --headerdigest none | CRC32`
ヘッダーのために CRC32 を有効または無効のどちらにするかを設定します。

`-p, --login-param`
1 つ以上のログインパラメータ設定を指定します。

注 – これらの値は、iSCSI ネットワーク内におけるこのパラメータの影響についての実際的な知識を備えた管理者だけが変更するようにしてください。

ログインパラメータは、iSCSI 標準 RFC 3720 から派生しています。 有効な値は次のとおりです。

`dataseqinorder`	`yes` または `no`
`defaulttime2retain`	0–3600
`defaulttime2wait`	0–3600
`firstburstlength`	512 から 2^{24-1}
`immediatedata`	`yes` または `no`
`initialr2t`	`yes` または `no`
`maxburstlength`	512 から 2^{24-1}
`datapduinorder`	`yes` または `no`
`maxoutstandingr2t`	1 から 65535
`maxrecvdataseglen`	512 から 2^{24-1}

`-T`, `--tunable-param` <<*tunable-prop*>=<*value*>, ...>
イニシエータノードが接続されているすべてのターゲットに対して 1 つ以上の調整可能なパラメータを指定します。

注 – 調整可能な値は、iSCSI ネットワーク内におけるこのパラメータの影響についての実際的な知識を備えた管理者だけが変更するようにしてください。

サポートされている *tunable-prop* オプションは次のとおりです。

`recv-login-rsp-timeout`
セッションログインの応答時間

`recv-login-rsp-timeout` オプションは、iSCSI イニシエータが iSCSI ターゲットからの iSCSI セッションログイン要求の応答を待つ時間を指定します。有効な値は 0 〜 60*60 であり、デフォルト値は 60 秒です。

`conn-login-max`
接続再試行の最大時間

`conn-login-max` オプションを指定すると、指定された時間ウィンドウ中に入出力タイムアウトまたは接続の失敗が発生した場合に、iSCSI イニシエータはターゲットへの接続を再確立できます。有効な値は 0 〜 60*60 であり、デフォルト値は 180 秒です。

`polling-login-delay`
ログイン再試行の時間間隔

`polling-login-delay` オプションは、入出力タイムアウトまたは接続の失敗が発生した場合の iSCSI イニシエータからターゲットへの各ログイン再試行の時間間隔を指定します。有効な値は 0 〜 60*60 であり、デフォルト値は 60 秒です。

remove サブコマンド

`remove` サブコマンドの構文は次のとおりです。

```
# iscsiadm remove direct_object
```

`remove` サブコマンドは、次の直接オブジェクトをサポートしています。

`discovery-address` *discovery-address*, ...
発見アドレスのリストからターゲットデバイスを削除します。発見アドレス (下に示す構文で指定される) は、SendTargets 発見セッションで使用される IP アドレスとポートの組み合わせです。この発見アプローチを使用すると、ターゲットデバイスはターゲットアドレスと、そのデバイスによって公開された各ターゲットのターゲット名をイニシエータに通知できます。発見アドレスによって公開されたいずれかのターゲットが現在マウントされているか、またはそのデバイス上にアクティブな I/O が存在する場合は、「論理ユニット使用中」のエラーが返され、操作は失敗します。 関連付けられたデバイスが使用されていない場合、それらのデバイスは削除されます。

discovery-address は、次の形式である必要があります。

<*IP address*>[:<*port*>]

この直接オブジェクトに関連付けられたオプションはありません。

`isns-server` *isns-server*, ...

iSNS サーバーアドレスのリストから iSNS サーバーを削除します。iSNS サーバーアドレス (下に示す構文で指定される) は、iSNS 発見セッションで使用される IP アドレスとポートの組み合わせです。iSNS 発見を使用すると、iSNS サーバーはイニシエータにポータルに関する情報や、そのイニシエータと同じ発見ドメインに属する各ターゲットの名前を提供できます。iSNS によって検出されたいずれかのターゲットが現在マウントされているか、またはそのデバイス上にアクティブな I/O が存在する場合は、「論理ユニット使用中」のエラーが返され、操作は失敗します。関連付けられたデバイスが使用されていない場合、それらのデバイスは削除されます。

isns-server は、次の形式である必要があります。

IP_address[:*port*]

この直接オブジェクトに関連付けられたオプションはありません。

`static-config` *static_target*, ...

静的に検出されたターゲットのリストからターゲットを削除します。削除されるターゲットが現在マウントされているか、またはそのデバイス上にアクティブな I/O が存在する場合は、「論理ユニット使用中」のエラーが返され、操作は失敗します。デバイスが使用されていない場合、そのデバイスは削除されます。

static_target は、次の形式である必要があります。

<*target-name*>,<*target-address*>[:*port-number*][,*tpgt*]

この直接オブジェクトに関連付けられたオプションはありません。

`target-param` *target-name*

target-name で指定されたターゲットを削除します。ターゲット名の形式は次のとおりです。

<*target-name*>

この直接オブジェクトに関連付けられたオプションはありません。Solaris I/O マルチパス機能 (以前は Sun StorEdge Traffic Manager [STMS] または MPxIO と呼ばれていた) が有効になっている場合の iSCSI ブートでは、ターゲットを削除することはできません。

ディスカバリ方式の正しい使用

単一のターゲットを、静的な発見方式と動的な発見方式の両方で検出されるようには構成しないでください。発見方式を重複して使用すると、iSCSI ターゲットデバイスとの通信時にパフォーマンスが低下する恐れがあります。

オプション

以下の汎用オプションがサポートされています。

-V, --version　バージョン情報を表示します。以降の引数の解釈を停止します。

-?, --help　ヘルプ情報を表示します。引数のない iscsiadm コマンドのあと、サブコマンドのあと、またはサブコマンドと直接オブジェクトの組み合わせのあとに使用できます。入力に適したヘルプ情報が表示されます。たとえば、次の行を入力したとします。

```
# iscsiadm modify initiator-node --help
```

...iscsiadm によって、そのサブコマンドと直接オブジェクトの組み合わせに使用可能なオプションが表示されます。

使用例

例 1　発見アドレスの追加

次のコマンドは、add サブコマンドを使用して発見アドレスを追加します。

```
# iscsiadm add discovery-address 10.0.0.1:3260 10.0.0.2:3260
```

例 2　静的なターゲットの追加

次のコマンドは、add サブコマンドを使用して静的なターゲットを追加します。

```
# iscsiadm add static-config \
iqn.1999-08.com.array:sn.01234567,10.0.0.1:3260
```

例 3　現在の発見設定の一覧表示

次のコマンドは、list サブコマンドを使用して現在の発見設定を一覧表示します。

```
# iscsiadm list discovery
        Discovery:
                Static: enabled
                Send Targets: disabled
                iSNS: enabled
```

例 4　詳細な発見出力の取得

次のコマンドは、list サブコマンドで -v オプションを使用して (使用した例と、使用しない例が 1 つずつ)、詳細な出力を取得します。

```
# iscsiadm list discovery-address
        Discovery Address: 10.0.0.1:3260
        Discovery Address: 10.0.0.2:3260

# iscsiadm list discovery-address -v 10.0.0.1:3260
        Discovery Address: 10.0.0.1:3260
                Target name: eui.210000203787d1f7
                        Target address:    10.0.0.1:3260
                Target name: eui.210000203787a693
                        Target address:    10.0.0.1:3260
```

例 5 イニシエータに関する情報の表示

次のコマンドは、list サブコマンドを使用してイニシエータに関する情報を表示します。

```
# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.company.central.interopv20-1
Initiator node alias: interopv20-1
        Login Parameters (Default/Configured):
                Header Digest: NONE/NONE
                Data Digest: NONE/NONE
        Authentication Type: CHAP
                CHAP Name: iqn.1986-03.com.company.central.interopv20-1
        RADIUS Server: NONE
        RADIUS access: disabled
        Tunable Parameters (Default/Configured):
                Session Login Response Time: 60/-
                Maximum Connection Retry Time: 180/-
                Login Retry Time Interval: 60/-
        Configured Sessions: 1
```

例 6 静的な設定情報の表示

次のコマンドは、list サブコマンドを使用して静的な設定に関する情報を表示します。

```
# iscsiadm list static-config
        Static target: eui.210000203787a693,10.0.0.1:3260
```

例 7 ターゲットの情報の表示

次のコマンドは、list サブコマンドをさまざまなオプションとともに使用してターゲットに関する情報を表示する方法を示しています。

```
# iscsiadm list target
Target: iqn.2004-05.com.abcStorage:Tgt-1
        Alias: -
        TPGT: 12288
        ISID: 4000002a0000
        Connections: 1# iscsiadm list target -v iqn.2004-05.com.abcStorage:Tgt-1
Target: iqn.2004-05.com.abcStorage:Tgt-1
        Alias: -
        TPGT: 12288
        ISID: 4000002a0000
        Connections: 1
                CID: 0
                  IP address (Local): 10.4.52.158:32803
                  IP address (Peer): 10.4.49.70:3260
                  Discovery Method: SendTargets
                  Login Parameters (Negotiated):
```

例 7 ターゲットの情報の表示 *(続き)*

```
                        Data Sequence In Order: yes
                        Data PDU In Order: yes
                        Default Time To Retain: 20
                        Default Time To Wait: 2
                        Error Recovery Level: 0
                        First Burst Length: 65536
                        Immediate Data: yes
                        Initial Ready To Transfer (R2T): yes
                        Max Burst Length: 262144
                        Max Outstanding R2T: 1
                        Max Receive Data Segment Length: 65536
                        Max Connections: 1
                        Header Digest: NONE
                        Data Digest: NONE
# iscsiadm list target -S iqn.2004-05.com.abcStorage:Tgt-1
Target: iqn.2004-05.com.abcStorage:Tgt-1
        Alias: -
        TPGT: 12288
        ISID: 4000002a0000
        Connections: 1
        LUN: 6
             Vendor:  ABCStorage
             Product: iSCSI Target
             OS Device Name: /dev/rdsk/c3t1d0s2
        LUN: 5
             Vendor:  ABCStorage
             Product: iSCSI Target
             OS Device Name: /dev/rdsk/c3t0d0s2
```

例 8 ターゲットのパラメータ情報の表示

次のコマンドは、`list` サブコマンドを使用して、特定のターゲットに関するターゲット情報を表示します。

```
# iscsiadm list target-param -v iqn.2004-05.com.abcStorage:Tgt-1
Target: iqn.2004-05.com.abcStorage:Tgt-1
        Alias: -
        Bi-directional Authentication: disabled
        Authentication Type: NONE
        Login Parameters (Default/Configured):
                Data Sequence In Order: yes/-
                Data PDU In Order: yes/-
                Default Time To Retain: 20/-
                Default Time To Wait: 2/-
                Error Recovery Level: 0/-
                First Burst Length: 65536/-
```

例 8 ターゲットのパラメータ情報の表示 (続き)

```
                Immediate Data: yes/-
                Initial Ready To Transfer (R2T): yes/-
                Max Burst Length: 262144/-
                Max Outstanding R2T: 1/-
                Max Receive Data Segment Length: 65536/-
                Max Connections: 1/-
                Header Digest: NONE/-
                Data Digest: NONE/-
        Tunable Parameters (Default/Configured):
                Session Login Response Time: 60/-
                Maximum Connection Retry Time: 180/-
                Login Retry Time Interval: 60/-
        Configured Sessions: 1
```

例 9 静的な発見方式の有効化

次のコマンドは、modify サブコマンドを使用して静的な発見方式を有効にします。

```
# iscsiadm modify discovery --static enable
```

例 10 RADIUS サーバーの IP アドレスの設定

次のコマンドは、modify サブコマンドを使用して、CHAP 認証に使用される RADIUS サーバーの IP アドレスを設定します。

```
# iscsiadm modify initiator --radius-server 10.0.0.1
```

例 11 イニシエータのノード名の設定

次のコマンドは、modify サブコマンドを使用してイニシエータノードのノード名を設定します。

```
# iscsiadm modify initiator-node -N iqn.2004-10.com.SUN.host-1
```

例 12 イニシエータノードの最大接続数の設定

次のコマンドは、modify サブコマンドを使用してイニシエータノードの最大接続数を設定します。これにより、1 つのセッションに複数の接続が存在できるようになります。

```
# iscsiadm modify initiator-node -m 3
```

例 13 ターゲットのパラメータの変更

次のコマンドは、modify サブコマンドを使用して、指定されたターゲットのターゲットパラメータの最大接続数を変更します。

```
# iscsiadm modify target-param -m 3 eui.210000203787a693
```

例 14 発見アドレスの削除

次のコマンドは、remove サブコマンドを使用して発見アドレスを削除します。

```
# iscsiadm remove discovery-address 10.0.0.1:3260
```

例 15 ターゲットのパラメータの削除

次のコマンドは、remove サブコマンドを使用して一連のターゲットパラメータを削除します。

```
# iscsiadm remove target-param eui.210000203787a693
```

例 16 最大接続数の変更

次のコマンドは、イニシエータのプロパティーで各セッションの最大接続数を変更します。変更された値は、すべてのセッションで、すべてのターゲットに使用されます。

```
# iscsiadm modify initiator-node --max-connections 4
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/iscsi/iscsi-initiator
インタフェースの安定性	確実

関連項目

attributes(5), iscsi(7D), scsi_vhci(7D)

『Oracle Solaris の管理: デバイスとファイルシステム』

名前

isns – iSNS (Internet Storage Name Service)

形式

`/usr/sbin/isns`

機能説明

iSNS (Internet Storage Name Service) は、IP ネットワーク内の iSCSI (インターネット SCSI) および iFCP (Internet Fibre Channel Protocol) デバイス向けの統合化された検出サービスを提供します。iSNS はクライアントサーバーメカニズムを使用します。サーバーはクライアントの構成情報を格納し、クライアントの要求に応じてその情報を提供します。サーバーとクライアント間のプロトコルについては、IETF RFC 4171 に記述されています。

このマニュアルページでは、Solaris iSNS サーバー実装の要約を説明します。現在の実装では iFCP デバイスはサポートされていません。

Solaris iSNS サーバーはデーモン `isns` として実装されています。これは、サービスクライアント要求への既知のポート 3205 にバインドします。このデーモンは、サービス管理機能 (`smf(5)`) により、障害管理リソース識別子 (FMRI) を使用して開始されます。

```
svc:/network/isns_server
```

`isns` を有効にするには、`svcadm(1M)` を使用します。サービスを有効にすると、オペレーティングシステムのブート時に常にサービスが自動的に開始および実行されます。サービスの状態は、`svcs(1)` コマンドを使用すると表示できます。

次に示すサービスのプロパティーは、`svccfg(1M)` を使用すると管理できます。デフォルト値は、RFC 4171 および実装の選択を通じて割り当てられます。

`data_store_location`
: 構成データストアの場所。デフォルトの場所は `/etc/isns/isnsdata.xml` です。

`ESI_retry_threshold_count`
: エンティティーステータス照会の再試行しきい値カウンタ。デフォルトのカウントは `3` です。

`Management_SCNs_Enabled`
: 管理状態変更通知を有効にするかどうかを決定するブール値。デフォルトは `yes` です。

`Authorized_Control_Nodes`
: 制御ノード名。

プロパティー値の変更後に、`svcadm(1M)` `refresh` を使用して `isns` を有効にし、新しい値を認識させる必要があります。`data_store_location` プロパティーを変更した場合は、`svcadm restart` コマンドを入力して変更を有効にする必要があります。

RFC 4171 には、デフォルト発見ドメイン、デフォルトドメインセット、および「デフォルト DD/DDS」設定が定義されています。これは、ユーザー定義の発見ドメイ

ンのいずれにも割り当てられていないクライアントを管理するためのものです。デフォルトの発見ドメインおよびドメインセットに対してサーバーが採用する動作を次に示します。

- 未割り当てのクライアントがデフォルト発見ドメインに追加されます。新たに登録されたクライアントまたは最後の発見ドメインメンバーシップから削除されたクライアントは、未割り当てのクライアントと見なされます。
- クライアントがユーザー定義の発見ドメインに割り当てられると、サーバーはデフォルト発見ドメインからクライアントを削除します。
- デフォルト発見ドメインセットは、管理者による有効化/無効化が可能です。これにより、デフォルト発見ドメイン内でのクライアントの発見を管理者が制御できます。
- 管理者がクライアントをデフォルト発見ドメインに追加することも、ユーザー定義の発見ドメインをデフォルト発見ドメインセットに追加することもできません。
- デフォルト発見ドメインセットのデフォルト状態は無効です。

isns サーバーでサポートされる rbac(5) 承認を使用すると、isns のアクティビティーを管理できます。これらの承認には、次の auth_attr(4) 特権が含まれています。

solaris.isnsmgr.write
: 発見ドメインやドメインセットの作成、発見ドメインセットの有効化/無効化、および発見ドメイン内の iSNS クライアントのグループや発見ドメインセット内の発見ドメインのグループの変更に必要です。

solaris.smf.manage.isns
: smf(5) を使用した isns サーバーの管理に必要です。

solaris.smf.value.isns
: isns に関連付けられている SMF サービスプロパティーの変更に必要です。

iSNS サーバー管理プロファイル (prof_attr(4) を参照) には、前述の承認がすべて含まれています。役割と承認の概要については、rbac(5) を参照してください。

オプション

isns デーモンによってサポートされているオプションはありません。

使用例

例 1　isns サーバーを起動する

次のコマンドは、isns サーバーを起動します。

```
# svcadm enable svc:/network/isns_server
```

例 2　isns サーバーを停止する

次のコマンドは、isns サーバーを停止します。

```
# svcadm disable svc:/network/isns_server
```

例 3　isns のプロパティーを変更する

次の一連のコマンドは、ESI_retry_threshold_count プロパティーの値を変更します。

```
# svccfg -s svc:/network/isns_server setprop \
config/ESI_retry_threshold_count = 6
# svcadm refresh svc:/network/isns_server
```

ファイル

/usr/sbin/isns
　iSNS デーモンバイナリ。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/isns
インタフェースの安定性	確実
標準	standards(5) を参照してください。

関連項目

svcs(1), isnsadm(1M), svcadm(1M), svccfg(1M), auth_attr(4), prof_attr(4), attributes(5), rbac(5), smf(5)

注意事項

サービスプロパティーの変更後に、サーバー (svcadm restart) を再起動することが強く推奨されています。これにより、サーバーは既存のクライアントにも新規のクライアントにも同一の設定を適用できます。

サーバーの管理に、RFC 4171 に記述されている制御ノードは不要です。制御ノードは、ローカルホスト上で isnsadm(1M) コマンドを使用して操作できます。たとえば、isnsadm を使って発見ドメインおよび発見ドメインセットを作成し、それにメンバーを追加して、発見ドメインと発見ドメインセットの関連付けを作成できます。

名前

isnsadm – iSNS (internet Storage Name Server) サーバーの管理

形式

`isnsadm` *options*

`isnsadm` *subcommand* [*subcommand_options*] [*operand*]

機能説明

`isnsadm` コマンドは、iSNS (Internet Storage Name Service) サーバーのコマンド行インタフェースです。`isnsadm` は、iSNS サーバー管理機能の 1 つを実行する個々のサブコマンドのセットで構成されています。各サブコマンドについては、それぞれの節を参照してください。

iSNS 構成を変更する操作には、`solaris.isnsmgr.write` 承認が必要です。isns(1M) を参照してください。読み取り操作では、このコマンドに特別な承認は不要です。

`isnsadm` には一般的なオプションのセットとサブコマンド固有オプションのセットが存在します。最初のカテゴリについては「オプション」を参照してください。2 つ目のカテゴリについては、各サブコマンドの説明を参照してください。

オプション

次のオプションがサポートされています。

`-?`、`--help`
コンテキストヘルプを表示します。以降の引数の解釈を停止します。

`-V`、`--version`
バージョン情報を表示します。以降の引数の解釈を停止します。

オペランド

1 つ以上の `isnsadm` サブコマンドにより、次のオペランドが使用されます。

iscsi-node-name
iSCSI ターゲットまたは iSCSI イニシエータの記号名。文字列の最大長は 223 文字。

discovery-domain-name
発見ドメインの記号名。文字列の最大長は 256 文字。

discovery-domain-set-name
発見ドメインセットの記号名。文字列の最大長は 256 文字。

サブコマンド

`isnsadm` コマンドは、次のサブコマンドをサポートします。

add-dd

`add-dd` サブコマンドは、発見ドメインを発見ドメインセットに追加します。

`add-dd` サブコマンドの構文は次のとおりです。

```
# isnsadm add-dd option discovery-domain-name, ...
```

`add-dd` には、次のオプションがあります。

`-s` *discovery-domain-set-name*
発見ドメインセットを指定します。

add-node

add-node サブコマンドは、指定された発見ドメインにノードを追加します。

add-node サブコマンドの構文は次のとおりです。

 # isnsadm add-node *option iscsi-node-name*,...

add-node には、次のオプションがあります。

-d、 --dd *discovery-domain-name*
発見ドメインを指定します。

create-dd

create-dd サブコマンドは、指定された名前の発見ドメインを作成します。

create-dd サブコマンドの構文は次のとおりです。

 # isnsadm create-dd *discovery-domain-name*, ...

create-dd にオプションはありません。

create-dd-set

create-dd-set サブコマンドは、指定された名前の発見ドメインセットを作成します。

create-dd-set サブコマンドの構文は次のとおりです。

 # isnsadm create-dd-set *discovery-domain-set-name*, ...

create-dd-set にオプションはありません。

delete-dd

delete-dd サブコマンドは、指定された名前の発見ドメインを削除します。

delete-dd サブコマンドの構文は次のとおりです。

 # isnsadm delete-dd *discovery-domain-name*, ...

delete-dd にオプションはありません。

delete-dd-set

delete-dd-set サブコマンドは、指定された名前の発見ドメインセットを削除します。

delete-dd-set サブコマンドの構文は次のとおりです。

 # isnsadm delete-dd-set *discovery-domain-set-name*, ...

delete-dd-set にオプションはありません。

disable-dd-set

disable-dd-set サブコマンドは、発見ドメインセットを無効にします。

disable-dd-set サブコマンドの構文は次のとおりです。

 # isnsadm disable-dd-set *discovery-domain-set-name*, ...

disable-dd-set にオプションはありません。

enable-dd-set

enable-dd-set サブコマンドは、発見ドメインセットを有効にします。

enable-dd-set サブコマンドの構文は次のとおりです。

```
# isnsadm enable-dd-set discovery-domain-set-name, ...
```

enable-dd-set にオプションはありません。

list-dd

list-dd サブコマンドは、発見ドメインに関する情報を表示します。オペランドが指定されない場合、現在 iSNS サーバーに存在するすべての発見ドメインの一覧を表示します。

list-dd サブコマンドの構文は次のとおりです。

```
# isnsadm list-dd [option] [discovery-domain-name, ...]
```

list-dd には、次のオプションを指定できます。

-v、--verbose
: 発見ドメインのメンバーの内容を表示します。

list-dd-set

list-dd-set サブコマンドは、iSNS サーバーに存在する有効な発見ドメインセットと無効なドメインセットの両方を一覧表示します。dd-set の登録は行われないことに注意してください。オペランドが指定されていない場合は、すべての発見ドメインセットの一覧が表示されます。

list-dd-set サブコマンドの構文は次のとおりです。

```
# isnsadm list-dd-set [option] [discovery-domain-set-name, ...]
```

list-dd-set には、次のオプションを設定できます。

-v、--verbose
: 発見ドメインセット内のすべての発見ドメインを表示します。

list-node

list-node サブコマンドは、iSNS サーバーに現在登録されているノード、または登録されておらず、デフォルト以外の発見ドメインに属するノードに関する情報を表示します。後者の場合、ノードにはタイプフィールド unknown が含まれています。オペランドが指定されていない場合、list-node は iSNS サーバーが認識するすべてのノードを表示します。

list-node サブコマンドの構文は次のとおりです。

```
# isnsadm list-node [options] [iscsi-node-name, ...]
```

list-node には、次のオプションを指定できます。

-t、--target
: リストをフィルタ処理して、iSCSI ターゲットノードだけを表示します。

-i、--initiator
: リストをフィルタ処理して、iSCSI イニシエータノードだけを表示します。

-v、--verbose
ノードの詳細を表示します。このオプションを使用しない場合、名前、別名、およびタイプ情報だけが表示されます。

modify-dd

modify-dd サブコマンドは、指定された発見ドメインの属性を変更します。

modify-dd サブコマンドの構文は次のとおりです。

 # isnsadm modify-dd *option discovery-domain-name*

modify-dd には、次のオプションがあります。

-n *discovery-domain-name*
既存の発見ドメインに適用する新しい発見ドメイン名を指定します。

modify-dd-set

modify-dd-set サブコマンドは、発見ドメインセットを変更します。

modify-dd-set サブコマンドの構文は次のとおりです。

 # isnsadm modify-dd-set *option discovery-domain-set-name*

modify-dd-set には、次のオプションがあります。

-n *discovery-domain-set-name*
既存の発見ドメインセットに適用する新しい発見ドメインセット名を指定します。

remove-dd

remove-dd サブコマンドは、指定された発見ドメインセットとの関連付けを削除します。

remove-dd サブコマンドの構文は次のとおりです。

 # isnsadm remove-dd *option discovery-domain-name*, **...**

remove-dd には、次のオプションがあります。

-s *discovery-domain-set-name*
発見ドメインを削除する発見ドメインセットを指定します。

remove-node

remove-node サブコマンドはノードを削除します。

remove-node サブコマンドの構文は次のとおりです。

 # isnsadm remove-node *option iscsi-node-name*, **...**

remove-node には、次のオプションがあります。

-d *discovery-domain-name*
ノードを削除する発見ドメインを指定します。

show-config

show-config サブコマンドは、iSNS サーバーの管理設定を表示します。設定は、サービス管理機能を使って変更できます (smf(5) を参照)。isns(1M) を参照してください。

show-config サブコマンドの構文は次のとおりです。

```
# isnsadm show-config
```

show-config にオプションはありません。

使用例

例 1 クライアントを表示する

次では、list-node サブコマンドを使用してクライアントを表示します。

```
# isnsadm list-node -v
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.45173FEA.engr
        Alias: STK5320_NAS
        Type: Target
        Network Entity: SE5310
        Portal: 172.20.57.95:3260
                Portal Group: 1
        Portal: 172.20.56.95:3260
                Portal Group: 1
        DD Name: Default
iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.acct
        Alias:
        Type: Target
        Network Entity: SE5310
        Portal: 172.20.57.95:3260
                Portal Group: 1
        Portal: 172.20.56.95:3260
                Portal Group: 1
        DD Name: Default
iSCSI Name: iqn.1986-03.com.sun:01:e00000000000.46fd8e2b
        Alias: host-x2100
        Type: Initiator
        Network Entity: iqn.1986-03.com.sun:01:e00000000000.46fd8e2b
        Portal: 172.20.236.123:58530
                Portal Group: 1
        DD Name: Default
```

例 2 発見ドメインを表示する

次では、list-dd サブコマンドを使用して発見ドメインを表示します。

```
# isnsadm list-dd -v
DD name: Default
        DD set(s): Default
    iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.45173FEA.engr
    iSCSI Name: iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.acct
```

例 2 発見ドメインを表示する (続き)

```
        iSCSI name: iqn.1986-03.com.sun:01:e00000000000.46fd8e2b
DD name: acct-dd
DD name: engineering-dd
```

例 3 ノードを追加する

次では、add-node サブコマンドを使用して発見ドメインにノードを追加することで、発見ドメインメンバーシップを作成します。

```
# isnsadm add-node -d engineering-dd \
iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.engr
```

例 4 ノードを削除する

次では、remove-node サブコマンドを使用して発見ドメインからノードを削除することにより、発見ドメインメンバーシップを削除します。

```
# isnsadm remove-node -d acct-dd \
iqn.1986-03.com.sun:01:000e0c9f10da.454F00A2.acct
```

例 5 発見ドメインセットを作成する

次では、create-dd-set サブコマンドを使用して発見ドメインセットを作成します。

```
# isnsadm create-dd-set operation-dd-set
```

例 6 発見ドメインセットを表示する

次では、list-dd-set サブコマンドを使用して発見ドメインセットを表示します。

```
# isnsadm list-dd-set -v
DD Set name: Default
        State: Disabled
        DD Name: Default
DD Set name: operation-dd-set
        State: Disabled
```

例 7 発見ドメインを追加する

次では、add-dd サブコマンドを使用して発見ドメインセットに発見ドメインを追加します。

```
# isnsadm add-dd -s operation-dd-set engineering-dd
```

例 8 発見ドメインセットを表示する

次では、list-dd-set を使用して発見ドメインセットの属性を表示します。

```
# isnsadm list-dd-set
DD Set name: Default
```

例 8 発見ドメインセットを表示する (続き)

```
        State: Disabled
        DD Name: Default
DD Set name: operation-dd-set
        State: Disabled
        DD Name: engineering-dd
```

例 9 発見ドメインセットを有効にする

次では、enable-dd-set サブコマンドを使用して発見ドメインセットを有効にします。

```
# isnsadm enable-dd-set Default
```

例 10 発見ドメインセットを無効にする

次では、disable-dd-set サブコマンドを使用して発見ドメインセットを無効にします。

```
# isnsadm disable-dd-set Default
```

例 11 管理設定を表示する

次では、show-config サブコマンドを使用して現在の管理設定を表示します。

```
# isnsadm show-config
        Data Store Location: /etc/isns/isnsdata.xml
        Entity Status Inquiry Non-Response Threshold: 3
        Management SCN Enabled: yes
        Authorized Control Node Names: -
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/isns
インタフェースの安定性	流動的

関連項目

iscsiadm(1M), isns(1M), attributes(5), smf(5)

注意事項

複数のオペランドを使ってサブコマンドが呼び出され、1 つ以上のオペランドが失敗したがすべてのオペランドが失敗したわけではない場合、isnsadm は部分障害の発生を示す一般的なメッセージおよび失敗したオペランドの一覧を表示します。特定のオペランドで発生したエラーを確認するには、失敗したオペランドを指定して同じサブコマンドを実行します。

名前

itadm – iSCSI ターゲットの管理

形式

```
itadm create-target [-a,--auth-method radius | chap | none | default]
     [-s,--chap-secret] [-S,--chap-secret-file path]
     [-u,--chap-user chap-user-name] [-n,--node-name target_node_name]
     [-l,--alias alias] [-t,--tpg tpg-name[,tpg-name]]

itadm modify-target [-a,--auth-method radius | chap | none | default]
     [-s,--chap-secret] [-S,--chap-secret-file path]
     [-u,--chap-user chap-user-name] [-n,--node-name new_target_node_name]
     [-l,--alias alias] [-t,--tpg tpg-name[,tpg-name]] target_node_name

itadm delete-target [-f,--force] target_node_name

itadm list-target [-v,--verbose] [target_node_name]

itadm create-tpg tpg_name IP-address[:port] [IP-address[:port]]...

itadm list-tpg [-v,--verbose] [tpg_name]

itadm delete-tpg  [-f,--force] tpg_name

itadm create-initiator [-s,--chap-secret] [-S,--chap-secret-file path]
     [-u,--chap-user chap-user-name] initiator_node_name

itadm modify-initiator [-s,--chap-secret] [-S,--chap-secret-file path]
     [-u,--chap-user chap-user-name] initiator_node_name

itadm list-initiator [-v,--verbose] initiator_node_name

itadm delete-initiator initiator_node_name

itadm modify-defaults [-a,--auth-method radius | chap | none]
     [-r,--radius-server IP-address[:port]] [-d,--radius-secret]
     [-D,--radius-secret-file path][-i,--isns enable | disable]
     [-I,--isns-server IP-address[:port][,IP-address[:port]]]

itadm list-defaults
```

機能説明

itadm コマンドは、SCSI ターゲットモードフレームワーク (SCSI Target Mode Framework) 内部の iSCSI (Internet SCSI) ターゲットノードを管理します。詳細については、stmfadm(1M) および libstmf(3LIB) を参照してください。 これにより、iSCSI イニシエータが iSCSI プロトコルを使用して STMF 論理ユニットにアクセスすることが可能になります。itadm は、iSCSI ターゲットノードのほかに、iSCSI ターゲットポータルグループおよび iSCSI イニシエータノードコンテキストという 2 つの管理対象オブジェクトクラスを管理します。

itadm は、独自のオプションとオペランドを備えたサブコマンドのセットとして実装されています。これらのサブコマンドについては、以降の節でサブコマンドごとに説明します。itadm は、サブコマンドのほかに、ユーティリティーの使用方法を示すヘルプコマンドも備えています。ヘルプコマンドを呼び出すには、-? オプションを使用します。

iSCSI ターゲットポータルグループ

iSCSI ターゲットネットワークポータルは、iSCSI ターゲットへの接続にイニシエータノードが使用可能な IP アドレスと TCP ポートです。これらのポータルの集まりをターゲットポータルグループ (TPG) と呼びます。TPG を使用すると、iSCSI ターゲットへのアクセスを制限できます。特定の iSCSI ターゲットを TPG にバインドするには、`itadm modify - t` コマンドを使用します。iSCSI リスナーは、TPG に属する各 IP アドレス上に作成され、iSCSI ターゲットへの接続を待機します。

TPG は、TPG の作成時に指定された一意の名前で識別されます。TPG の作成時に 2 - 65535 の範囲の数値「ターゲットポータルグループタグ」が自動的に生成されます。ターゲットポータルグループタグ 1 は、「デフォルト」のターゲットポータルグループ用に予約されています。これは、ターゲットに明示的なターゲットポータルグループが設定されていない場合に使用されます。デフォルト TPG のポータルは、ポート 3260 上のすべてのネットワークインタフェースからの要求と一致します。

iSCSI イニシエータノードコンテキスト

チャレンジハンドシェーク認証プロトコル (CHAP) による特定の操作 (認証など) には、リモート iSCSI イニシエータノードに関連付けられたパラメータが必要です。これらのパラメータは、iSCSI イニシエータノードコンテキストと関連付けられています。iSCSI イニシエータノードコンテキストは、IQN または EUI 形式で記述されたイニシエータノード名で識別されます (RFC 3720 を参照)。例:

```
iqn.1986-03.com.sun:01:e00000000000.47d55444
eui.02004567A425678D
```

IP アドレスの指定

多くの `itadm` サブコマンドでは、1 つ以上の IP アドレスを指定する必要があります。ポート番号の指定は任意です。IPv4 の場合は、標準のドット区切り 10 進数表記を使用します。IPv6 の場合は、アドレスを角括弧で囲みます。次に指定の例を示します。

```
IPv4:  10.2.4.1
       10.2.4.1:3260
IPv6:  [1080:0:0:0:8:800:200C:417A]
       [1080:0:0:0:8:800:200C:417A]:3260
```

サブコマンド

次に、`itadm` のサブコマンドとそのオプションを示します。

itadm create-target

```
itadm create-target [-a,--auth-method radius | chap | none | default]
     [-s,--chap-secret]
     [-S,--chap-secret-file path] [-u,--chap-user chap-user-name]
     [-n,--node-name target_node_name] [-l,--alias alias]
     [-t,--tpg tpg-name[,tpg-name,...]]
```

オプションを指定して iSCSI ターゲットを作成します。オプションは、次のとおりです。

`-a、--auth-method radius | chap | none | default`
: ターゲットに使用する認証方法を指定します。有効な値は、`radius`、`chap`、および `none` です。`chap` は、このターゲットに接続するイニシエータをチャレンジハンドシェーク認証プロトコル (CHAP) を使用して認証する必要があることを示し

ます。radius は、イニシエータが CHAP によっても認証されるが、必須の認証パラメータは中央の RADIUS サーバーから取得されることを示します (radius-server および radius-secret オプションを参照)。 none は、ターゲットへの接続に認証が不要であることを示します。default は、ターゲットがこのプロパティーのグローバル設定を使用することを示します(modify-defaults サブコマンドを参照)。

-s、--chap-secret
相互 CHAP 認証時に送信する CHAP シークレット。このプロパティーにデフォルトはありません。最大長は 255 文字、最低限必要な長さは 12 文字です。

-S、--chap-secret-file *path*
-s オプションで指定された CHAP シークレットを含む一時ファイルへのパス。

-u、--chap-user *chap-user-name*
相互 CHAP 認証で使用するターゲットの CHAP ユーザー名を指定します。この値はターゲットのみに許可され、グローバルに設定することはできません。また、この値を使用できるのは、相互 CHAP 認証を使用するようにイニシエータノードが構成されている場合だけです。値が指定されていない場合、ターゲットのノード名がユーザー名として使用されます。iscsiadm(1M) を参照してください。

-n、--node-name *target_node_name*
iSCSI ターゲットノードは、IQN または EUI 形式で記述されたターゲットノード名で識別されます (RFC 3720 を参照)。このオプションにより、その名前が設定されます。

-l、--alias *alias*
ターゲットノードに関連付けられた代替識別子。この識別子は一意でなくてもかまいません。

-t、--tpg *tpg-name*[,*tpg-name*,...]
ターゲットポータルグループ (TPG) 識別子のリスト。このリストには、イニシエータから特定のターゲットへのアクセスに使用可能な TPG やキーワード default を指定します。default が指定された場合、ターゲットはデフォルトのポータル INADDR_ANY:3260 を使用します。

itadm modify-target

```
itadm modify-target [-a,--auth-method radius | chap | none | default]
    [-s,--chap-secret] [-S,--chap-secret-file path]
    [-u,--chap-user chap-user-name] [-n,--node-name new_tgt_node_name]
    [-l,--alias alias] [-t,--tpg tpg-name[,tpg-name]] target_node_name
```

指定されたオプションに従って iSCSI ターゲットを変更します。オプションは、次のとおりです。

-a、--auth-method radius | chap | none | default
前述の create-target サブコマンドで説明したとおりです。

-s、--chap-secret
前述の create-target サブコマンドで説明したとおりです。

-S、--chap-secret-file *path*
前述の create-target サブコマンドで説明したとおりです。

-u、--chap-user *chap-user-name*
前述の create-target サブコマンドで説明したとおりです。明示的に設定した CHAP ユーザー名を削除するには、-u none を使用します。

-n、--node-name *target_node_name*
ターゲットの名前を変更します。前述の create-target サブコマンドの -n の説明を参照してください。

-l、--alias *alias*
前述の create-target サブコマンドで説明したとおりです。明示的に設定した別名を削除するには、-l none を使用します。

-t、--tpg *tpg-name*[,*tpg-name*,...]
前述の create-target サブコマンドで説明したとおりです。

itadm list-target

itadm list-target [-v,--verbose] [*target_node_name*]

構成済みのターゲットに関する情報を一覧表示します。*target_node_name* が指定されている場合は、そのターゲットの情報のみを一覧表示します。オプションは、次のとおりです。

-v、--verbose
冗長モード。

itadm delete-target

itadm delete-target [-f,--force] *target_node_name*

target_node_name で指定されたターゲットを削除します。ターゲットを削除するには、オフラインにする必要があります。オプションは、次のとおりです。

-f、--force
ターゲットのオンライン状態が続く場合、このオプションはターゲットを削除する前にオフライン化を試みます。

itadm create-tpg

itadm create-tpg *tpg_name IP-address*[:*port*]...

指定されたポータルで構成される iSCSI ターゲットのポータルグループを作成して、それに識別子 *tpg_name* を割り当てます。各ポータルは、IP アドレスとポートのペアです。IPv4 ポータルのアドレスは、ドット表記で指定します (例: 172.31.255.255)。IPv6 ポータルアドレスは、角括弧で囲む必要があります。

このサブコマンドにオプションはありません。

itadm list-tpg

itadm list-tpg [-v,--verbose] [*tpg_name*]

構成済みのターゲットポータルグループに関する情報を一覧表示します。*tpg_name* が指定されている場合、*tpg_name* に関連付けられているターゲットポータルグループの情報だけを一覧表示します。オプションは、次のとおりです。

`-v、 --verbose`
冗長モード。

itadm delete-tpg

`itadm delete-tpg [-f,--force]` *tpg_name*

tpg_name と関連付けられているターゲットポータルグループを削除します。オプションは、次のとおりです。

`-f、 --force`
TPG がいずれかのターゲットと関連付けられている場合、このオプションが指定されていないかぎり、削除要求は拒否されます。

itadm create-initiator

`itadm create-initiator [-s,--chap-secret] [-S,--chap-secret-file` *path*`]`
`[-u,--chap-user` *chap-user-name*`]` *initiator_node_name*

initiator_node_name という名前のリモートイニシエータと関連付けられているパラメータを構成します。オプションは、次のとおりです。

`-s、 --chap-secret`
前述の `create-target` サブコマンドで説明したとおりです。

`-S、 --chap-secret-file` *path*
前述の `create-target` サブコマンドで説明したとおりです。

`-u、 --chap-user` *chap-user-name*
CHAP 認証で使用するイニシエータの CHAP ユーザー名を指定します。値が指定されていない場合、イニシエータのノード名がユーザー名として使用されます。

itadm modify-initiator

`itadm modify-initiator [-s,--chap-secret] [-S,--chap-secret-file` *path*`]`
`[-u,--chap-user` *chap-user-name*`]` *initiator_node_name*

initiator_node_name という名前のリモートイニシエータと関連付けられているパラメータを変更します。オプションは、次のとおりです。

`-s、 --chap-secret`
前述の `create-target` サブコマンドで説明したとおりです。

`-S、 --chap-secret-file` *path*
前述の `create-target` サブコマンドで説明したとおりです。

`-u、 --chap-user` *chap-user-name*
CHAP 認証で使用するイニシエータの CHAP ユーザー名を指定します。値が指定されていない場合、イニシエータのノード名がユーザー名として使用されます。

itadm delete-initiator

`itadm delete-initiator` *initiator_node_name*

initiator_node_name という名前のリモートイニシエータと関連付けられているパラメータを削除します。このサブコマンドにオプションはありません。

itadm list-initiator

```
itadm list-initiator [-v,--verbose] initiator_node_name
```

initiator_node_name という名前のイニシエータと関連付けられているパラメータを一覧表示します。オプションは、次のとおりです。

`-v`、`--verbose`
冗長モード。

itadm modify-defaults

```
itadm modify-defaults [-a,--auth-method radius | chap | none]
    [-r,--radius-server IP-address[:port]] [-d,--radius-secret]
    [-D,--radius-secret-file path][-i,--isns enable | disable]
    [-I,--isns-server IP-address[:port][,IP-address[:port]]]
```

デフォルトのパラメータを変更します。オプションは、次のとおりです。

`-a`、`--auth-method radius | chap | none`
すべてのターゲットで使用するデフォルトの認証方法を指定します。有効な値は、`radius`、`chap`、および `none` です。`chap` は、このターゲットに接続するイニシエータをチャレンジハンドシェーク認証プロトコル (CHAP) を使用して認証する必要があることを示します。`radius` は、イニシエータが CHAP によっても認証されるが、必須の認証パラメータは中央の RADIUS サーバーから取得されることを示します(`--radius-server` および `--radius-secret` オプションを参照)。 `none` は、ターゲットへの接続に認証が不要であることを示します。`create-target` および `modify-target` サブコマンドの `-a` オプションを使用すると、このグローバル設定をターゲットごとに上書きできます。

`-d`、`--radius-secret`
集中管理される CHAP 認証の RADIUS 共有シークレット。

`-D`、`--radius-secret-file` *path*
`-d` オプションで指定された CHAP シークレットを含む一時ファイルへのパス。

`-i`、`--sns enable | disable`
定義済み iSNS (iSCSI Name Service) サーバーのセットにターゲットを登録するかどうかを指定します。

`-I`、`--isns-server` *IP-address*[:*port*][,*IP-address*[:*port*],...]
個別のターゲットと関連付けられた `isns` オプションが設定されている場合に、iSCSI ターゲットノードを登録する iSNS サーバーのリストを定義します。最大 8 台の iSNS サーバーを指定できます。すべての iSNS サーバーを削除するには、`-I none` を使用します。

`-r`、`--radius-server` *IP-address*[:*port*]
集中管理される CHAP 認証で使用する RADIUS サーバーの IP アドレスを指定します。

itadm list-defaults

```
itadm list-defaults
```

デフォルトプロパティーの情報を一覧表示します。このサブコマンドにオプションはありません。

使用例

例 1 ターゲットを作成する

次のコマンドは、ターゲットを作成します。

```
# itadm create-target
Target iqn.1986-03.com.sun:02:72e1b181-7bce-c0e6-851e-ec0d8cf14b7a
successfully created
```

例 2 指定した名前のターゲットを作成する

次のコマンドは、特定の IQN を使ってターゲットを作成します。

```
# itadm create-target -n eui.20387ab8943ef7548
or:
# itadm create-target \
-n iqn.1986-03.com.sun:02:a9a366f8-cc2b-f291-840948c7f29e
```

例 3 名前を変更する

次のコマンドは、既存のターゲットの IQN を変更します。

```
# itadm modify-target -n eui.20387ab8943ef7548 \
iqn.1986-03.com.sun:02:a9a366f8-909b-cc2b-f291-840948c7f29e
```

例 4 CHAP 認証を設定する

次のコマンドは、デフォルトの CHAP ユーザー名を使用してターゲットの CHAP 認証を設定します。

```
# itadm modify-initiator -s iqn.1986-03.com.sun:01:e00000000000.47d55444
Enter CHAP secret: *********
Re-enter secret: *********

# itadm modify-target -a chap eui.20387ab8943ef7548
```

例 5 ターゲットポータルグループを作成する

次のコマンドは、TPG 2 のアドレスにポート 8000 を使用して、2 つのターゲットポータルグループ A および B を作成します。

```
# itadm create-tpg A 192.168.0.1 192.168.0.2
# itadm create-tpg B 192.168.0.2:8000 192.168.0.2:8000
```

例 6 TPG の使用に合わせてターゲットを構成する

次のコマンドは、TPG A および B の使用に合わせてターゲットを構成します。

```
# itadm modify-target -t A,B eui.20387ab8943ef7548
```

例 7 特定のターゲット用の RADIUS 認証を設定する

次のコマンドは、特定のターゲット用の RADIUS 認証を設定します。

例 7 特定のターゲット用の RADIUS 認証を設定する (続き)

```
# itadm modify-defaults -r 192.168.10.1 -d
Enter RADIUS secret: *********
Re-enter secret: *********

# itadm modify-target -a radius eui.20387ab8943ef7548
```

例 8 すべてのターゲット用の RADIUS 認証を設定する

次のコマンドは、すべてのターゲット用の RADIUS 認証を設定します。

```
# itadm modify-defaults -d -r 192.168.10.1 -a radius
Enter RADIUS secret: *********
Re-enter secret: *********
```

このコマンドは、すべてのターゲットが -a default を使って作成されたことを前提としています。

例 9 デフォルトプロパティーを一覧表示する

次のコマンドは、デフォルトプロパティーを一覧表示します。

```
# itadm list-defaults
iSCSI Target Default Properties:

alias:          none
auth:           none
radiusserver:   none
radiussecret:   unset
isns:           disabled
isnsserver:     2.3.4.5,4.5.6.7
```

例 10 ターゲットを一覧表示する

次のコマンドは、ターゲットを一覧表示します。

```
# itadm list-target
TARGET NAME                                          STATE    SESSIONS
iqn.1986-03.com.sun:02:72e1b181-7bce-c0e6-851e-ec0d8cf14b7a  online   0
iqn.1986-03.com.sun:02:2cb0c526-c05a-e279-e396-a367006f4227  online   0
iqn.1986-03.com.sun:02:d14125bb-1c9d-c28d-97b0-f89259b642f3  online   0
iqn.1986-03.com.sun:02:03ff9fc5-794a-e9b4-a081-bb82917c292a  online   0
```

例 11 ターゲットを一覧表示する (冗長)

次のコマンドは、冗長オプションを指定してターゲットを一覧表示します。

```
# itadm list-target -v
TARGET NAME                                          STATE    SESSIONS
iqn.1986-03.com.sun:02:d23e68d8-2d79-c988-98e7-a6361689d33c  online   0
```

例 11 ターゲットを一覧表示する (冗長) (続き)

```
        alias:                  -
        auth:                   none (defaults)
        targetchapuser:         -
        targetchapsecret:       unset
        tpg-tags:               default
iqn.1986-03.com.sun:02:94ec46d4-c8e1-6993-ef03-ffc1dcd66606  online   1
        alias:                  -
        auth:                   chap
        targetchapuser:         -
        targetchapsecret:       unset
        tpg-tags:               nge1_ipv4 = 3
```

例 12 特定のターゲットを一覧表示する

次のコマンドは、冗長オプションを指定してターゲットを一覧表示します。

```
# itadm list-target -v \
iqn.1986-03.com.sun:02:2cb0c526-c05a-e279-e396-a367006f4227
TARGET NAME                                          STATE    SESSIONS
iqn.1986-03.com.sun:02:2cb0c526-c05a-e279-e396-a367006f4227  online   1
        alias:                  -
        auth:                   chap
        targetchapuser:         -
        targetchapsecret:       unset
        tpg-tags:               nge1_ipv4 = 3
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/iscsi/iscsi-target
インタフェースの安定性	確実

関連項目

iscsiadm(1M), stmfadm(1M), libstmf(3LIB), attributes(5)

名前

itu – convert packages to Driver Update format and patch Solaris install media for Install Time Update

形式

```
itu makedu -r solaris_release [-v] [-f] [-d output_dir] [-o iso_file]
     [-l iso_label] package [package...]

itu updatemedia -R media_root [-v] [-f] [-o iso_file]
     [-l iso_label] pkg_or_patch [pkg_or_patch...]

itu makeiso -o iso_file [-v] [-l iso_label] media_root
```

機能説明

The `itu` utility converts driver packages to Driver Update (DU) format and patches a Solaris install media with driver packages and patches for Install Time Update (ITU). `itu` has three subcommands: `makedu`, `updatemedia` and `makeiso`.

オプション

The following options are supported:

`-d` *output_dir*
: Directory where the Driver Update directory is to be created.

`-f`
: If *output_dir*/DU or *iso_file* already exists, remove it without asking first.

`-l` *iso_label*
: Label/volume name of the ISO image (if `-o` option is specified).

`-o` *iso_file*
: Path of the ISO image file to be created. For subcommands `updatemedia` and `makeiso`, it will be a bootable ISO image. This option must be specified for subcommand `makeiso`.

`-R` *media_root*
: Top-level directory of on-disk image of Solaris installation media. This option must be specified for subcommand `updatemedia`.

`-r` *solaris_release*
: Solaris release number for which the Driver Update is intended. It takes the form of the output of `uname -r`, for example, `5.10`. This option must be specified for subcommand `makedu`.

`-v`
: Verbose. Multiple `-v` options increase verbosity.

サブコマンド

The `itu` subcommands are described as follows.

makedu

The `makedu` subcommand takes one or more driver packages as input and converts them to DU format. At the beginning of an interactive Solaris installation session, these driver updates can be applied to the running kernel, which will then also automatically apply them to the newly installed Solaris at the end of the installation process.

The `-r` option is required to specify the Solaris release number for which the driver updates apply. The *solaris_release* option argument takes the form `uname -r` output, for example, `5.10` or `5.11`.

If the `-d` option is specified, the resulting DU directory tree is placed in the directory *output_dir*.

If the `-o` option is specified, a (non-bootable) ISO image of the DU directory tree is written in the file *iso_file*. This ISO image can be burned onto a CD/DVD using `cdrw(1)` or `cdrecord(1)` (not a SunOS man page).

At least one of `-d` and `-o` option must be specified. If both are specified, then both an ISO image and a directory tree are generated.

updatemedia

The `updatemedia` subcommand takes a list of driver packages and patches as input and applies them to the miniroot of a Solaris install media. It also places them in a subdirectory called `ITUs` under the Solaris install media's top-level directory:

media_root`/ITUs`

When booting a system from the updated media, the patches and packages will be part of the booted Solaris image. They will also be applied to the target system being installed at the end of the installation process.

The `-R` option must be entered on the command line to specify the Solaris install media. Note that the install media must be on a location that is writable by `itu`.

If the `-o` option is specified, a bootable ISO image of the patched install media is also created in the file *iso_file*. The ISO image can then be burned onto a CD or DVD.

makeiso

The `makeiso` subcommand runs `mkisofs(8)` to create a bootable Solaris ISO image of the Solaris install media *media_root* and writes it to the file *iso_file*. The ISO image file can then be burned onto a CD or DVD with utilities such as `cdrw(1)` or `cdrecord(1)`. (Note that `mkisofs(8)` and `cdrecord(1)` are not SunOS man pages.)

注意 – The Solaris install media *media-root* must contain the file `boot/grub/stage2_eltorito`, which will be written to the media boot sectors. This file will be modified with some boot information, thus it has to be writable. If necessary, first save a copy, prior to running this subcommand.

オペランド

The following operands are supported:

package [*package*...]
: One or more driver packages.

pkg_or_patch [*pkg_or_patch*...]
: One or more patches or packages.

media_root
: The top-level directory of a Solaris install media.

使用例

例 1 Creating a DU CD/DVD

The following commands create a Driver Update CD/DVD containing the packages SAMPLEpkg1 and SAMPLEpkg2.

```
# itu makedu -r 5.10 -o my.iso SAMPLEpkg1 SAMPLEpkg2
# cdrw -i my.iso
```

例 2 Patching the Solaris Install Media

The following command patches the Solaris install media in /export/s10u1 with patch /opt/patches/123456-07 and driver package /opt/pkgs/MYdriver. The command also creates a bootable ISO image with ISO label "MyS10U1" in the file /tmp/dvd.iso.

```
# /usr/bin/itu updatemedia -R /export/s10u1 -o /tmp/dvd.iso -l MyS10U1 \
       /opt/patches/123456-07 /opt/pkgs/MYdriver
```

例 3 Creating a Bootable ISO Image

The following commands create the bootable ISO image mydvd.iso of the Solaris install image /export/solaris-10u1 with ISO label "Special-S10".

```
# /usr/bin/itu makeiso -o mydvd.iso -l "Special-S10" \
     /export/solaris-10u1
# cdrw -i mydvd.iso
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

cdrw(1), pkgadd(1M), attributes(5)

mkisofs(8), (/usr/share/man/man8/mkisofs.8), in the SUNWfsman package (not a SunOS man page)

名前

k5srvutil – host key table (keytab) manipulation utility

形式

`/usr/sbin/k5srvutil` *operation* `[-ik] [-f` *filename*`]`

機能説明

The `k5srvutil` command allows a system manager to list or change keys currently in his keytab or to add new keys to the keytab.

The operand *operation* must be one of the following:

`list` Lists the keys in a keytab, showing version number and principal name.

`change` Changes all the keys in the keytab to new randomly-generated keys, updating the keys in the Kerberos server's database to match those by using the `kadmin` protocol. If a key's version number does not match the version number stored in the Kerberos server's database, the operation fails. The old keys are retained so that existing tickets continue to work. If the `-i` flag is specified, `k5srvutil` prompts for `yes` or `no` before changing each key. If the `-k` option is used, the old and new keys are displayed.

`delold` Deletes keys that are not the most recent version from the keytab. This operation should be used at some point after a change operation to remove old keys. If the `-i` flag is specified, `k5srvutil` asks the user whether the old keys associated with each principal should be removed.

`delete` Deletes particular keys in the keytab, interactively prompting for each key.

In all cases, the default keytab file is `/etc/krb5.keytab` file unless this is overridden by the `-f` option.

`k5srvutil` uses the `kadmin(1M)` program to edit the keytab in place. However, old keys are retained, so they are available in case of failure.

オプション

The following options are supported:

`-f` *filename* Specify a keytab file other than the default file, `/etc/krb5.keytab`.

`-i` Prompts user before changing keys when using the `change` or `delold` operands.

`-k` Displays old and new keys when using the `change` operand.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目 ktutil(1), kadmin(1M), attributes(5)

名前

kadb – a kernel debugger

形式

SPARC

```
ok boot device_specifier kadb [-d] [boot-flags]
```

x86

```
select (b)oot or (i)nterpreter: b kadb [-d] [boot-flags]
```

機能説明

kadb, an interactive kernel debugger, has been replaced by kmdb(1). For backwards compatibility, the methods used to load kadb will be interpreted as requests to load kmdb(1). Unlike with the compatibility link from adb(1) to mdb(1), kmdb(1) will always load in its native user interface mode, regardless of the name used to load it.

kmdb(1) is based on mdb(1), and thus shares mdb's user interface style and feature set. The mdb(1) man page describes the features and operation of mdb. The kmdb(1) man page describes the differences between mdb and kmdb. This man page describes the major changes and incompatibilities between kadb and kmdb.

Consult the Solaris Modular Debugger Guide for a detailed description of both mdb and kmdb.

Major changes

This section briefly lists the major differences between kadb and kmdb. It is not intended to be exhaustive.

Debugger Loading and Unloading	kmdb(1) may be loaded at boot, as with kadb. It may also be loaded after boot, thus allowing for kernel debugging and execution control without requiring a system reboot. If kmdb(1) is loaded after boot, it may be unloaded.
mdb Feature Set	The features introduced by mdb(1), including access to kernel type data, debugger commands (dcmds), debugger modules (dmods), and enhanced execution control facilities, are available under kmdb(1). Support for changing the representative CPU (:x) is available for both SPARC and x86. Furthermore, full execution-control facilities are available after the representative CPU has been changed.

Significant Incompatibilities

This section lists the significant features that have changed incompatibly between kadb and kmdb(1). It is not intended to be exhaustive. All kmdb(1) commands referenced here are fully described in the kmdb(1) man page. A description as well as examples can be found in the Solaris Modular Debugger Guide.

Deferred Breakpoints	The kadb-style “module#symbol:b” syntax is not supported under kmdb(1). Instead, use “::bp module‘symbol”.
Watchpoints	The ::wp dcmd is the preferred way to set watchpoint with kmdb. Various options are available to control the type of watchpoint set, including -p for physical watchpoints

(SPARC only), and -i for I/O port watchpoints (x86 only). $l is not supported, therefore, the watchpoint size must be specified for each watchpoint created.

Access to I/O Ports (x86 only) — The commands used to access I/O ports under kadb have been replaced with the ::in and ::out dcmds. These two dcmds allow both read and write of all I/O port sizes supported by kadb.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/processor

関連項目

adb(1), mdb(1), kmdb(1), attributes(5)

Solaris Modular Debugger Guide

名前

kadmin, kadmin.local – Kerberos database administration program

形式

```
/usr/sbin/kadmin [-r realm] [-p principal] [-q query]
 [-s admin_server [:port]] [ [-c credential_cache]
 | [-k [-t keytab] | [-n]] | [-w password]] [-x db_args]...
```

```
/usr/sbin/kadmin.local [-r realm] [-p principal]
 [-q query] [-d dbname] [-e "enc:salt..."] [-m] [-D]
```

機能説明

`kadmin` and `kadmin.local` are interactive command-line interfaces to the Kerberos V5 administration system. They provide for the maintenance of Kerberos principals, policies, and service key tables (keytabs). `kadmin` and `kadmin.local` provide identical functionality; the difference is that `kadmin.local` can run only on the master KDC and does not use Kerberos authentication.

Except as explicitly noted otherwise, this man page uses `kadmin` to refer to both versions.

By default, both versions of `kadmin` attempt to determine your user name and perform operations on behalf of your "*username*/*admin*" instance. Operations performed are subject to privileges granted or denied to this user instance by the Kerberos ACL file (see `kadm5.acl(4)`). You may perform administration as another user instance by using the `-p` option.

The remote version, `kadmin`, uses Kerberos authentication and an encrypted RPC to operate securely from anywhere on the network. It normally prompts for a password and authenticates the user to the Kerberos administration server, `kadmind`, whose service principal is `kadmin/`*fqdn*. Some options specific to the remote version permit the password prompt to be bypassed. The `-c` option searches the named credentials cache for a valid ticket for the `kadmin/`*fqdn* service and uses it to authenticate the user to the Kerberos admin server without a password. The `-k` option searches a keytab for a credential to authenticate to the `kadmin/`*fqdn* service, and again no password is collected. If `kadmin` has collected a password, it requests a `kadmin/`*fqdn* Kerberos service ticket from the KDC, and uses that service ticket to interact with `kadmind`.

The local version, `kadmin.local`, must be run with an effective UID of root, and normally uses a key from the `/var/krb5/.k5.`*realm* stash file (see `kdb5_util(1M)`) to decrypt information from the database rather than prompting for a password. The `-m` option will bypass the `.k5.`*realm* stash file and prompt for the master password.

オプション

The following options are supported:

`-c` *credentials_cache*
: Search *credentials_cache* for a service ticket for the `kadmin/`*fqdn* service; it can be acquired with the `kinit(1)` program. If this option is not specified, `kadmin` requests a new service ticket from the KDC, and stores it in its own temporary credentials cache.

`-d` *dbname*
: Specify a non-standard database name. [Local only]

`-D`
: Turn on debug mode. [Local only]

`-e` *"enc:salt ..."*
Specify a different encryption type and/or key salt. [Local only]

`-k` [`-t` *keytab*]
Use the default keytab (`-k`) or a specific keytab (`-t` *keytab*) to decrypt the KDC response instead of prompting for a password. In this case, the default principal will be `host/hostname`. This is primarily used for keytab maintenance.

`-m`
Accept the database master password from the keyboard rather than using the `/var/krb5/.k5.`*realm* stash file. [Local only]

`-n`
Requests anonymous processing. Two types of anonymous principals are supported. For fully anonymous Kerberos, configure `pkinit` on the KDC and configure `pkinit_anchors` in the client's `krb5.conf`. Then use the `-n` option with a principal of the form `@`*REALM* (an empty principal name followed by the at-sign and a realm name). If permitted by the KDC, an anonymous ticket will be returned. A second form of anonymous tickets is supported; these realm-exposed tickets hide the identity of the client but not the client's realm. For this mode, use `kinit -n` with a normal principal name. If supported by the KDC, the principal (but not realm) will be replaced by the anonymous principal. As of release 1.8, the MIT Kerberos KDC supports only fully anonymous operation.

`-p` *principal*
Authenticate *principal* to the `kadmin/`*fqdn* service. Otherwise, `kadmin` will append `/admin` to the primary principal name of the default credentials cache, the value of the `USER` environment variable, or the username as obtained with `getpwuid`, in that order of preference.

`-q` *query*
Pass *query* directly to `kadmin`, which will perform *query* and then exit. This can be useful for writing scripts.

`-r` *realm*
Use *realm* as the default database realm.

`-s` *admin_server*[*:port*]
Administer the specified *admin* server at the specified port number (*port*). This can be useful in administering a realm not known to your client.

`-w` *password*
Use *password* instead of prompting for one. Note that placing the password for a Kerberos principal with administration access into a shell script can be dangerous if unauthorized users gain read access to the script or can read arguments of this command through `ps(1)`.

`-x` *db_args*
Pass database-specific arguments to `kadmin`. Supported arguments are for LDAP and the `Berkeley-db2` plug-in. These arguments are:

`binddn=`*binddn*
LDAP simple bind DN for authorization on the directory server. Overrides the `ldap_kadmind_dn` parameter setting in `krb5.conf`(4).

`bindpwd=`*bindpwd*
Bind password.

`dbname=`*name*
For the `Berkeley-db2` plug-in, specifies a name for the Kerberos database.

`nconns=`*num*
Maximum number of server connections.

`port=`*num*
Directory server connection port.

Commands

`list_requests`
Lists all the commands available for `kadmin`. Aliased by `lr` and `?`.

`get_privs`
Lists the current Kerberos administration privileges (ACLs) for the principal that is currently running `kadmin`. The privileges are based on the `/etc/krb5/kadm5.acl` file on the master KDC. Aliased by `getprivs`.

`add_principal` [*options*] *newprinc*
Creates a new principal, *newprinc*, prompting twice for a password. If the `-policy` option is not specified and a policy named `default` exists, then the `default` policy is assigned to the principal; note that the assignment of the `default` policy occurs automatically only when a principal is first created, so the `default` policy must already exist for the assignment to occur. The automatic assignment of the `default` policy can be suppressed with the `-clearpolicy` option. This command requires the `add` privilege. Aliased by `addprinc` and `ank`. The options are:

`-expire` *expdate*
Expiration date of the principal. See the `Time Formats` section for the valid absolute time formats that you can specify for *expdate*.

`-pwexpire` *pwexpdate*
Password expiration date. See the `Time Formats` section for the valid absolute time formats that you can specify for *pwexpdate*.

`-maxlife` *maxlife*
Maximum ticket life for the principal. See the `Time Formats` section for the valid time duration formats that you can specify for *maxlife*.

`-maxrenewlife` *maxrenewlife*
Maximum renewable life of tickets for the principal. See the `Time Formats` section for the valid time duration formats that you can specify for *maxrenewlife*.

`-kvno` *kvno*
Explicitly set the key version number.

`-policy` *policy*
: Policy used by the principal. If both the `-policy` and `-clearpolicy` options are not specified, the `default` policy is used if it exists; otherwise, the principal will have no policy. Also note that the password and principal name must be different when you add a new principal with a specific policy or the `default` policy.

`-clearpolicy`
: `-clearpolicy` prevents the `default` policy from being assigned when `-policy` is not specified. This option has no effect if the `default` policy does not exist.

{–|+}`allow_postdated`
: `-allow_postdated` prohibits the principal from obtaining postdated tickets. (Sets the `KRB5_KDB_DISALLOW_POSTDATED` flag.) `+allow_postdated` clears this flag.

{–|+}`allow_forwardable`
: `-allow_forwardable` prohibits the principal from obtaining forwardable tickets. (Sets the `KRB5_KDB_DISALLOW_FORWARDABLE` flag.) `+allow_forwardable` clears this flag.

{–|+}`allow_renewable`
: `-allow_renewable` prohibits the principal from obtaining renewable tickets. (Sets the `KRB5_KDB_DISALLOW_RENEWABLE` flag.) `+allow_renewable` clears this flag.

{–|+}`allow_proxiable`
: `-allow_proxiable` prohibits the principal from obtaining proxiable tickets. (Sets the `KRB5_KDB_DISALLOW_PROXIABLE` flag.) `+allow_proxiable` clears this flag.

{–|+}`allow_dup_skey`
: `-allow_dup_skey` disables user-to-user authentication for the principal by prohibiting this principal from obtaining a session key for another user. (Sets the `KRB5_KDB_DISALLOW_DUP_SKEY` flag.) `+allow_dup_skey` clears this flag.

{–|+}`requires_preauth`
: `+requires_preauth` requires the principal to preauthenticate before being allowed to `kinit`. (Sets the `KRB5_KDB_REQUIRES_PRE_AUTH` flag.) `-requires_preauth` clears this flag.

{–|+}`requires_hwauth`
: `+requires_hwauth` requires the principal to preauthenticate using a hardware device before being allowed to kinit. (Sets the `KRB5_KDB_REQUIRES_HW_AUTH` flag.) `-requires_hwauth` clears this flag.

{–|+}`allow_svr`
: `-allow_svr` prohibits the issuance of service tickets for the principal. (Sets the `KRB5_KDB_DISALLOW_SVR` flag.) `+allow_svr` clears this flag.

{–|+}`allow_tgs_req`
: –`allow_tgs_req` specifies that a Ticket-Granting Service (TGS) request for a service ticket for the principal is not permitted. This option is useless for most things.

+allow_tgs_req clears this flag. The default is +allow_tgs_req. In effect, –allow_tgs_req sets the KRB5_KDB_DISALLOW_TGT_BASED flag on the principal in the database.

{–|+}allow_tix
–allow_tix forbids the issuance of any tickets for the principal. +allow_tix clears this flag. The default is +allow_tix. In effect, –allow_tix sets the KRB5_KDB_DISALLOW_ALL_TIX flag on the principal in the database.

{–|+}needchange
+needchange sets a flag in attributes field to force a password change; –needchange clears it. The default is –needchange. In effect, +needchange sets the KRB5_KDB_REQUIRES_PWCHANGE flag on the principal in the database.

{–|+}password_changing_service
+password_changing_service sets a flag in the attributes field marking this as a password change service principal (useless for most things).
–password_changing_service clears the flag. This flag intentionally has a long name. The default is –password_changing_service. In effect, +password_changing_service sets the KRB5_KDB_PWCHANGE_SERVICE flag on the principal in the database.

{-|+}ok_as_delegate
+ok_as_delegate sets the OK-AS-DELEGATE flag on tickets issued for use with this principal as the service, which clients may use as a hint that credentials can and should be delegated when authenticating to the service. (Sets the KRB5_KDB_OK_AS_DELEGATE flag.) -ok_as_delegate clears this flag.

{-|+}ok_to_auth_as_delegate
+ok_to_auth_as_delegate sets the service to allow the use of S4U2Self.
-ok_to_auth_as_delegate clears this flag.

-randkey
Sets the key of the principal to a random value.

-pw *password*
Sets the key of the principal to the specified string and does not prompt for a password. Note that using this option in a shell script can be dangerous if unauthorized users gain read access to the script.

-e “enc:salt ...”
Override the list of enctype:salttype pairs given in kdc.conf(4) for setting the key of the principal. The quotes are necessary if there are multiple enctype:salttype pairs. One key for each similar enctype and same salttype will be created and the first one listed will be used. For example, in a list of two similar enctypes with the same salt, “des-cbc-crc:normal des-cbc-md5:normal”, one key will be created and it will be of type des-cbc-crc:normal.

`-unlock`
Unlocks the principal so that it can successfully authenticate. If the principal had previously been locked due to reaching `maxfailure` in `failurecountinterval` time then the principal will be locked for `lockoutduration` time.

Example:

```
kadmin: addprinc tlyu/admin
WARNING: no policy specified for "tlyu/admin@ACME.COM";
defaulting to no policy.
Enter password for principal tlyu/admin@ACME.COM:
Re-enter password for principal tlyu/admin@ACME.COM:
Principal "tlyu/admin@ACME.COM" created.
kadmin:
```

Errors:

`KADM5_AUTH_ADD` (requires `add` privilege)

`KADM5_BAD_MASK` (should not happen)

`KADM5_DUP` (principal exists already)

`KADM5_UNK_POLICY` (policy does not exist)

`KADM5_PASS_Q_*` (password quality violations)

`delete_principal` [`-force`] *principal*
Deletes the specified principal from the database. This command prompts for deletion, unless the `-force` option is given. This command requires the `delete` privilege. Aliased by `delprinc`.

Example:

```
kadmin: delprinc mwm_user
Are you sure you want to delete the principal
"mwm_user@ACME.COM"? (yes/no): yes
Principal "mwm_user@ACME.COM" deleted.
Make sure that you have removed this principal from
all kadmind ACLs before reusing.
kadmin:
```

Errors:

`KADM5_AUTH_DELETE` (requires `delete` privilege)

`KADM5_UNK_PRINC` (principal does not exist)

`modify_principal` [*options*] *principal*
Modifies the specified principal, changing the fields as specified. The options are as above for `add_principal`, except that password changing is forbidden by this command. In addition, the option `-clearpolicy` will clear the current policy of a principal. This command requires the `modify` privilege. Aliased by `modprinc`.

Errors:

KADM5_AUTH_MODIFY (requires modify privilege)

KADM5_UNK_PRINC (principal does not exist)

KADM5_UNK_POLICY (policy does not exist)

KADM5_BAD_MASK (should not happen)

change_password [*options*] *principal*

Changes the password of *principal*. Prompts for a new password if neither -randkey or -pw is specified. Requires the changepw privilege, or that the principal that is running the program to be the same as the one changed. Aliased by cpw. The following options are available:

-randkey

Sets the key of the principal to a random value.

-pw *password*

Sets the password to the specified string. Not recommended.

-e "enc:salt ..."

Override the list of enctype:salttype pairs given in kdc.conf(4) for setting the key of the principal. The quotes are necessary if there are multiple enctype:salttype pairs. For each key, the first matching similar enctype and same salttype in the list will be used to set the new key(s).

-keepold

Keeps the previous kvno's keys around. There is no easy way to delete the old keys, and this flag is usually not necessary except perhaps for TGS keys as it will allow existing valid TGTs to continue to work.

Example:

```
kadmin: cpw systest
Enter password for principal systest@ACME.COM:
Re-enter password for principal systest@ACME.COM:
Password for systest@ACME.COM changed.
kadmin:
```

Errors:

KADM5_AUTH_MODIFY (requires the modify privilege)

KADM5_UNK_PRINC (principal does not exist)

KADM5_PASS_Q_* (password policy violation errors)

KADM5_PASS_REUSE (password is in principal's password history)

KADM5_PASS_TOOSOON (current password minimum life not expired)

`get_principal` [`-terse`] *principal*

Gets the attributes of *principal*. Requires the `inquire` privilege, or that the principal that is running the program to be the same as the one being listed. With the `-terse` option, outputs fields as quoted tab-separated strings. Aliased by `getprinc`.

Examples:

```
kadmin: getprinc tlyu/admin
Principal: tlyu/admin@ACME.COM
Expiration date: [never]
Last password change: Thu Jan 03 12:17:46 CET 2008
Password expiration date: [none]
Maximum ticket life: 24855 days 03:14:07
Maximum renewable life: 24855 days 03:14:07
Last modified: Thu Jan 03 12:17:46 CET 2008 (root/admin@ACME.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 5
Key: vno 2, AES-256 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 2, AES-128 CTS mode with 96-bit SHA-1 HMAC, no salt
Key: vno 2, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 2, ArcFour with HMAC/md5, no salt
Key: vno 2, DES cbc mode with RSA-MD5, no salt
Attributes: REQUIRES_PRE_AUTH
Policy: [none]
kadmin: getprinc -terse tlyu/admin
"tlyu/admin@ACME.COM"   0       1199359066      0       2147483647
"root/admin@ACME.COM"   1199359066      128     2       0       "[none]"       21474836
47      0       0       0       5       1       2       18      0       1       2
17      0       1       2       16      0       1       2       23      0       12
       3       0
kadmin:
```

Errors:

`KADM5_AUTH_GET` (requires the get [inquire] privilege)

`KADM5_UNK_PRINC` (principal does not exist)

`list_principals` [*expression*]

Retrieves all or some principal names. *expression* is a shell-style glob expression that can contain the wild-card characters ?, *, and []'s. All principal names matching the expression are printed. If no expression is provided, all principal names are printed. If the expression does not contain an “@” character, an “@” character followed by the local realm is appended to the expression. Requires the `list` privilege. Aliased by `listprincs`, `get_principals`, and `getprincs`.

Examples:

```
kadmin: listprincs test*
test3@ACME.COM
test2@ACME.COM
test1@ACME.COM
testuser@ACME.COM
kadmin:
```

`add_policy` [*options*] *policy*

Adds the named policy to the policy database. Requires the `add` privilege. Aliased by `addpol`. The following options are available:

`-maxfailure` *maxnumber*

sets the maximum number of failures before the principal is locked after authentication failures in *failurecountinterval* time.

`-failurecountinterval` *failuretime*

sets the time after which the authentication failure count is reset 0. See the "Time Formats" section, below, for the valid time duration formats that you can specify for *failuretime*.

`-lockoutduration` *lockouttime*

sets the time in which the principal is locked from authenticating if *maxfailure* authentication failures occur within *failurecountinterval* time. See the "Time Formats" section, below, for the valid time duration formats that you can specify for *lockouttime*.

`-maxlife` *maxlife*

sets the maximum lifetime of a password. See the `Time Formats` section for the valid time duration formats that you can specify for *maxlife*.

`-minlife` *minlife*

sets the minimum lifetime of a password. See the `Time Formats` section for the valid time duration formats that you can specify for *minlife*.

`-minlength` *length*

sets the minimum length of a password.

`-minclasses` *number*

sets the minimum number of character classes allowed in a password. The valid values are:

`1`

only letters (himom)

`2`

both letters and numbers (hi2mom)

`3`

letters, numbers, and punctuation (hi2mom!)

`-history` *number*
sets the number of past keys kept for a principal.

Errors:
`KADM5_AUTH_ADD` (requires the `add` privilege)

`KADM5_DUP` (policy already exists)

`delete_policy` `[-force]` *policy*
Deletes the named policy. Unless the `-force` option is specified, prompts for confirmation before deletion. The command will fail if the policy is in use by any principals. Requires the `delete` privilege. Aliased by `delpol`.

Example:

```
kadmin: del_policy guests
Are you sure you want to delete the
policy "guests"? (yes/no): yes
Policy "guests" deleted.
kadmin:
```

Errors:
`KADM5_AUTH_DELETE` (requires the delete privilege)

`KADM5_UNK_POLICY` (policy does not exist)

`KADM5_POLICY_REF` (reference count on policy is not zero)

`modify_policy` `[`*options*`]` *policy*
Modifies the named policy. Options are as above for `add_policy`. Requires the `modify` privilege. Aliased by `modpol`.

Errors:
`KADM5_AUTH_MODIFY` (requires the `modify` privilege)

`KADM5_UNK_POLICY` (policy does not exist)

`get_policy` `[-terse]` *policy*
Displays the values of the named policy. Requires the `inquire` privilege. With the `-terse` flag, outputs the fields as quoted strings separated by tabs. Aliased by `getpol`.

Examples:

```
kadmin: get_policy admin
Policy: admin
Maximum password life: 180 days 00:00:00
Minimum password life: 00:00:00
Minimum password length: 6
Minimum number of password character classes: 2
Number of old keys kept: 5
Reference count: 17
```

```
Maximum password failures before lockout: 3
Password failure count reset interval: 180
Password lockout duration: 60
kadmin: get_policy -terse admin
admin admin    15552000    0    6    2    5    17 3 180 60
kadmin:
```

Errors:

KADM5_AUTH_GET (requires the get privilege)

KADM5_UNK_POLICY (policy does not exist)

list_policies [*expression*]

Retrieves all or some policy names. *expression* is a shell-style glob expression that can contain the wild-card characters ?, *, and []'s. All policy names matching the expression are printed. If no expression is provided, all existing policy names are printed. Requires the list privilege. Aliased by listpols, get_policies, and getpols.

Examples:

```
kadmin: listpols
test-pol dict-only once-a-min test-pol-nopw
kadmin: listpols t*
test-pol test-pol-nopw kadmin:
```

ktadd [-k *keytab*] [-q] [-e *keysaltlist*] [-norandkey] [[*principal* | -glob *princ-exp*]] [...]

Adds a principal or all principals matching *princ-exp* to a keytab. It randomizes each principal's key in the process, to prevent a compromised admin account from reading out all of the keys from the database. However, kadmin.local has the -norandkey option, which leaves the keys and their version numbers unchanged, similar to the Kerberos V4 ext_srvtab command. That allows users to continue to use the passwords they know to login normally, while simultaneously allowing scripts to login to the same account using a keytab. There is no significant security risk added since kadmin.local must be run by root on the KDC anyway.

ktadd requires the inquire and changepw privileges. An entry for each of the principal's unique encryption types is added, ignoring multiple keys with the same encryption type but different salt types. If the -k argument is not specified, the default keytab file, /etc/krb5/krb5.keytab, is used.

The "-e *enctype*:salt" option overrides the list of *enctypes* given in krb5.conf(4), in the permitted_enctypes parameter. If "-e *enctype*:salt" is not used and permitted_enctypes is not defined in krb5.conf(4), a key for each *enctype* supported by the system on which kadmin is run will be created and added to the keytab. Restricting the *enctypes* of keys in the keytab is useful when the system for which keys are being created does not support the same set of *enctypes* as the KDC. Note that ktadd modifies the *enctype* of the keys in the principal database as well.

If the -q option is specified, less status information is displayed. Aliased by xst. The -glob option requires the list privilege. Also, note that if you use -glob to create a keytab, you need to remove /etc/krb5/kadm5.keytab and create it again if you want to use -p */admin with kadmin.

princ-exp
princ-exp follows the same rules described for the list_principals command.

Example:

```
kadmin: ktadd -k /tmp/new-keytab nfs/chicago
Entry for principal nfs/chicago with kvno 2,
encryption type DES-CBC-CRC added to keytab
WRFILE:/tmp/new-keytab.
kadmin:
```

ktremove [-k *keytab*] [-q] *principal* [*kvno* | all | old]
Removes entries for the specified principal from a keytab. Requires no privileges, since this does not require database access. If all is specified, all entries for that principal are removed; if old is specified, all entries for that principal except those with the highest kvno are removed. Otherwise, the value specified is parsed as an integer, and all entries whose kvno match that integer are removed. If the -k argument is not specified, the default keytab file, /etc/krb5/krb5.keytab, is used. If the -q option is specified, less status information is displayed. Aliased by ktrem.

Example:

```
kadmin: ktremove -k /tmp/new-keytab nfs/chicago
Entry for principal nfs/chicago with kvno 2
removed from keytab
WRFILE:/tmp/new-keytab.
kadmin:
```

quit
Quits kadmin. Aliased by exit and q.

Time Formats

Various commands in kadmin can take a variety of time formats, specifying time durations or absolute times. The kadmin option variables *maxrenewlife*, *maxlife*, and *minlife* are time durations, whereas *expdate* and *pwexpdate* are absolute times.

Examples:

```
kadmin: modprinc -expire "12/31 7pm" jdb
kadmin: modprinc -maxrenewlife "2 fortnight" jdb
kadmin: modprinc -pwexpire "this sunday" jdb
kadmin: modprinc -expire never jdb
kadmin: modprinc -maxlife "7:00:00pm tomorrow" jdb
```

Note that times which do not have the “ago” specifier default to being absolute times, unless they appear in a field where a duration is expected. In that case, the time specifier will be interpreted as relative. Specifying “ago” in a duration can result in unexpected behavior.

The following time formats and units can be combined to specify a time. The time and date format examples are based on the date and time of July 2, 1999, 1:35:30 p.m.

Time Format	Examples
hh[:*mm*][:*ss*][am/pm/a.m./p.m.]	1p.m., 1:35, 1:35:30pm

Variable	Description
hh	hour (12-hour clock, leading zero permitted but not required)
mm	minutes
ss	seconds

Date Format	Examples
mm/dd[/*yy*]	07/02, 07/02/99
yyyy-mm-dd	1999-07-02
dd-*month-yyyy*	02-July-1999
month [,*yyyy*]	Jul 02, July 02,1999
dd *month*[*yyyy*]	02 JULY, 02 july 1999

Variable Description	
dd	day
mm	month
yy	year within century (00-38 is 2000 to 2038; 70-99 is 1970 to 1999)
yyyy	year including century
month	locale's full or abbreviated month name

Time Units	Examples
[+|- #] year	“-2 year”
[+|- #] month	“2 months”
[+|- #] fortnight	

[+|- #] week	
[+|- #] day	
[+|- #] hour	
[+|- #] minute	
[+|- #] min	
[+|- #] second	
[+|- #] sec	
tomorrow	
yesterday	
today	
now	
this	“this year”
last	“last saturday”
next	“next month”
sunday	
monday	
tuesday	
wednesday	
thursday	
friday	
saturday	
never	

You can also use the following time modifiers: `first`, `second`, `third`, `fourth`, `fifth`, `sixth`, `seventh`, `eighth`, `ninth`, `tenth`, `eleventh`, `twelfth`, and `ago`.

環境変数

See `environ(5)` for descriptions of the following environment variables that affect the execution of `kadmin`:

`PAGER`
: The command to use as a filter for paging output. This can also be used to specify options. The default is `more(1)`.

ファイル

`/var/krb5/principal`
Kerberos principal database.

`/var/krb5/principal.ulog`
The update log file for incremental propagation.

`/var/krb5/principal.kadm5`
Kerberos administrative database. Contains policy information.

`/var/krb5/principal.kadm5.lock`
Lock file for the Kerberos administrative database. This file works backwards from most other lock files (that is, `kadmin` will exit with an error if this file does *not* exist).

`/var/krb5/kadm5.dict`
Dictionary of strings explicitly disallowed as passwords.

`/etc/krb5/kadm5.acl`
List of principals and their `kadmin` administrative privileges.

`/etc/krb5/kadm5.keytab`
Keytab for `kadmind` principals: `kadmin/`*fqdn*, `changepw/`*fqdn*, and `kadmin/changepw`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

`kpasswd(1)`, `more(1)`, `gkadmin(1M)`, `kadmind(1M)`, `kdb5_util(1M)`, `kdb5_ldap_util(1M)`, `kproplog(1M)`, `kadm5.acl(4)`, `kdc.conf(4)`, `krb5.conf(4)`, `attributes(5)`, `environ(5)`, `kerberos(5)`, `krb5envvar(5)`

History

The `kadmin` program was originally written by Tom Yu at MIT, as an interface to the OpenVision Kerberos administration program.

診断

The `kadmin` command is currently incompatible with the MIT `kadmind` daemon interface, so you cannot use this command to administer an MIT-based Kerberos database. However, clients running the Solaris implementation of Kerberos can still use an MIT-based KDC.

名前

kadmind – Kerberos administration daemon

形式

```
/usr/lib/krb5/kadmind [-nofork] [-m] [-port port-number] [-r realm]
     [-x db_args] [-P pid_file]
```

機能説明

kadmind runs on the master key distribution center (KDC), which stores the principal and policy databases. kadmind accepts remote requests to administer the information in these databases. Remote requests are sent, for example, by kpasswd(1), gkadmin(1M), and kadmin(1M) commands, all of which are clients of kadmind. When you install a KDC, kadmind is set up in the init scripts to start automatically when the KDC is rebooted.

kadmind requires a number of configuration files to be set up for it to work:

/etc/krb5/kdc.conf
: The KDC configuration file contains configuration information for the KDC and the Kerberos administration system. kadmind understands a number of configuration variables (called relations) in this file, some of which are mandatory and some of which are optional. In particular, kadmind uses the acl_file, dict_file, admin_keytab, and kadmind_port relations in the [*realms*] section. Refer to the kdc.conf(4) man page for information regarding the format of the KDC configuration file.

/etc/krb5/kadm5.keytab
: kadmind requires a keytab (key table) containing correct entries for the kadmin/*fqdn*, kadmin/changepw and kadmin/changepw principals for every realm that kadmind answers requests. The keytab can be created with the kadmin(1M) or kdb5_util(1M) command. The location of the keytab is determined by the admin_keytab relation in the kdc.conf(4) file.

/etc/krb5/kadm5.acl
: kadmind uses an ACL (access control list) to determine which principals are allowed to perform Kerberos administration actions. The path of the ACL file is determined by the acl_file relation in the kdc.conf file. See kdc.conf(4). For information regarding the format of the ACL file, refer to kadm5.acl(4).

  The kadmind daemon will need to be restarted to reread the kadm5.acl file after it has been modified. You can do this, as root, with the following command:

  ```
  # svcadm restart svc:/network/security/kadmin:default
  ```

After kadmind begins running, it puts itself in the background and disassociates itself from its controlling terminal.

kadmind can be configured for incremental database propagation. Incremental propagation allows slave KDC servers to receive principal and policy updates incrementally instead of receiving full dumps of the database. These settings can be changed in the kdc.conf(4) file:

sunw_dbprop_enable = [true | false]
: Enable or disable incremental database propagation. Default is false.

`sunw_dbprop_master_ulogsize = N`
: Specifies the maximum amount of log entries available for incremental propagation to the slave KDC servers. The maximum value that this can be is 2500 entries. Default value is 1000 entries.

The `kiprop/`<*hostname*>`@`<*REALM*> principal must exist in the master's `kadm5.keytab` file to enable the slave to authenticate incremental propagation from the master. In the principal syntax above, <*hostname*> is the master KDC's host name and <*REALM*> is the realm in which the master KDC resides.

Kerberos client machines can automatically migrate Unix users to the default Kerberos realm specified in the local `krb5.conf(4)`, if the user does not have a valid kerberos account already. You achieve this by using the `pam_krb5_migrate(5)` service module for the service in question. The Kerberos service principal used by the client machine attempting the migration needs to be validated using the `u` privilege in `kadm5.acl(4)`. When using the `u` privilege, `kadmind` validates user passwords using PAM, specifically using a `PAM_SERVICE` name of `k5migrate` by calling `pam_authenticate(3PAM)` and `pam_acct_mgmt(3PAM)`.

A suitable PAM stack configuration example for `k5migrate` would look like:

```
k5migrate        auth    required        pam_unix_auth.so.1
k5migrate        account required        pam_unix_account.so.1
```

オプション

The following options are supported:

`-nofork`
: Specifies that `kadmind` does not put itself in the background and does not disassociate itself from the terminal. In normal operation, you should use the default behavior, which is to allow the daemon to put itself in the background.

`-m`
: Specifies that the master database password should be retrieved from the keyboard rather than from the stash file. When using `-m`, the `kadmind` daemon receives the password prior to putting itself in the background. If used in combination with the `-d` option, you must explicitly place the daemon in the background.

`-port` *port-number*
: Specifies the port on which the `kadmind` daemon listens for connections. The default is controlled by the `kadmind_port` relation in the `kdc.conf(4)` file.

`-P` *pid_file*
: Specifies the file to which the PID of `kadmind` process should be written to after it starts up. This can be used to identify whether `kadmind` is still running and to allow `init` scripts to stop the correct process.

`-r` *realm*
: Specifies the default realm that `kadmind` serves. If *realm* is not specified, the default *realm* of the host is used. `kadmind` answers requests for any realm that exists in the local KDC database and for which the appropriate principals are in its `keytab`.

`-x` *db_args*
Pass database-specific arguments to `kadmind`. Supported arguments are for LDAP and the `Berkeley-db2` plug-in. These arguments are:

`binddn=`*binddn*
LDAP simple bind DN for authorization on the directory server. Overrides the `ldap_kadmind_dn` parameter setting in `krb5.conf(4)`.

`bindpwd=`*bindpwd*
Bind password.

`dbname=`*name*
For the `Berkeley-db2` plug-in, specifies a name for the Kerberos database.

`nconns=`*num*
Maximum number of server connections.

`port=`*num*
Directory server connection port.

ファイル

`/var/krb5/principal`
Kerberos principal database.

`/var/krb5/principal.ulog`
The update log file for incremental propagation.

`/var/krb5/principal.kadm5`
Kerberos administrative database containing policy information.

`/var/krb5/principal.kadm5.lock`
Kerberos administrative database lock file. This file works backwards from most other lock files (that is, `kadmin` exits with an error if this file does not exist).

`/var/krb5/kadm5.dict`
Dictionary of strings explicitly disallowed as passwords.

`/etc/krb5/kadm5.acl`
List of principals and their `kadmin` administrative privileges.

`/etc/krb5/kadm5.keytab`
Keytab for `kadmin` principals: `kadmin/`*fqdn*, `changepw/`*fqdn*, and `kadmin/changepw`.

`/etc/krb5/kdc.conf`
KDC configuration information.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

kpasswd(1), svcs(1), gkadmin(1M), kadmin(1M), kadmin(1M), kdb5_util(1M), kdb5_ldap_util(1M), kproplog(1M), svcadm(1M), pam_acct_mgmt(3PAM), pam_authenticate(3PAM), kadm5.acl(4), kdc.conf(4), krb5.conf(4), attributes(5), kerberos(5), krb5envvar(5), pam_krb5_migrate(5), smf(5)

注意事項

The Kerberos administration daemon (kadmind) is now compliant with the change-password standard mentioned in RFC 3244, which means it can now handle change-password requests from non-Solaris Kerberos clients.

The kadmind service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/security/kadmin
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

The -d and -p arguments are made obsolete with the -nofork and -port arguments, respectively. The -d and -p arguments might be removed in a future release of the Solaris operating system.

名前

kcfd – kernel-level cryptographic framework daemon

形式

```
kcfd
```

機能説明

The `kcfd` daemon helps in managing CPU usage by cryptographic operations performed in software by kernel threads. The system utilization associated with these threads is charged to the `kcfd` process. It also does module verification for kernel cryptographic modules.

Only a privileged user can run this daemon.

The `kcfd` daemon is automatically invoked in run level 1, after `/usr` is mounted. A previously invoked `kcfd` daemon that is still running must be stopped before invoking another `kcfd` command.

Manually starting and restarting `kcfd` is not recommended. If it is necessary to do so, use the `cryptoadm(1M)` `start` and `stop` subcommands.

終了ステータス

The following exit values are returned:

`0` Daemon started successfully.

`> 1` Daemon failed to start.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`cryptoadm(1M)`, `attributes(5)`

名前

kclient – set up a machine as a Kerberos client

形式

```
/usr/sbin/kclient [-n] [-R realm] [-k kdc] [-a adminuser]
    [-c filepath] [-d dnsarg] [-f fqdn_list] [-h logical_host_name]
    [-k kdc_list] [-m master_kdc] [-p profile] [-s pam_service]
    [-T kdc_vendor]
```

機能説明

By specifying the various command options, you can use the `kclient` utility to:

- Configure a machine as a Kerberos client for a specified realm and for KDC by setting up `krb5.conf(4)`.
- Add the Kerberos host principal to the local host's `keytab` file (`/etc/krb5/krb5.keytab`).
- Set up the machine to do kerberized NFS.
- Bring over a master `krb5.conf` copy from a specified pathname.
- Setup a machine to do server and/or host/domain name-to-realm mapping lookups by means of DNS.
- Configure a Kerberos client to use an MS Active Directory server. This generates a `keytab` file with the Kerberos client's service keys populated.
- Setup a Kerberos client that has no service keys. This is useful when the client does not require service keys, because the client does not wish to host a service that uses Kerberos for security.
- Configure a Kerberos client that is part of a cluster. This option requires the logical host name of the cluster so that the proper service keys are created and populated in the client's `keytab` file.
- Setup a Kerberos client to join an environment that consists of Kerberos servers that are non-Solaris and non-MS Active Directory servers.
- Configure `pam.conf(4)` to use Kerberos authentication for specified services.
- Configure the client as a simple NTP broadcast/multicast client.
- Specify custom domain/host name-to-realm name mappings.
- Setup the Kerberos client to use multiple KDC servers.

The `kclient` utility needs to be run on the client machine with root permission and can be run either interactively or non-interactively. In the non-interactive mode, the user feeds in the required inputs by means of a profile, command-line options, or a combination of profile and command-line options. The user is prompted for “required” parameter values (`realm` and `adminuser`), if found missing in the non-interactive run. The interactive mode is invoked when the utility is run without any command-line arguments.

Both the interactive and non-interactive forms of `kclient` can add the `host/fqdn` entry to the local host's `keytab` file. They also can require the user to enter the password for the administrative user requested, to obtain the Kerberos Ticket Granting Ticket (TGT) for `adminuser`. The `host/fqdn`, `nfs/fqdn`, and `root/fqdn` principals can be added to the KDC database (if not already present) before their possible addition to the local host's `keytab`.

The `kclient` utility assumes that the local host has been setup for DNS and requires the presence of a valid `resolv.conf(4)`. Also, `kclient` can fail if the localhost time is not synchronized with that of the KDC. For Kerberos to function the localhost time must be within five minutes of that of the KDC. It is advised that both systems run some form of time synchronization protocol, such as the Network Time Protocol (NTP). See the `ntpd` man page, delivered in the `SUNWntpu` package (not a SunOS man page).

オプション

The non-interactive mode supports the following options:

`-n`
: Set up the machine for kerberized NFS. This involves making changes to `krb5*` security flavors in `nfssec.conf(4)`. This option will also add `nfs/fqdn` and `root/fqdn` entries to the local host's `keytab` file if the `-K` option has not been specified.

`-R` [*realm*]
: Specifies the Kerberos realm.

`-k` *kdc_list*
: The `-k` option specifies the KDC host names for the Kerberos client. *kdc_list* is a comma-separated list of KDCs. If the `-m` option is not used, it is assumed that the first (or only) host in *kdc_list* is the master KDC host name. Note that the list specified is used verbatim. This is helpful when specifying non-fully qualified KDC host names that can be canonicalized by DNS.

`-a` [*adminuser*]
: Specifies the Kerberos administrative user.

`-T` *kdc_vendor*
: Configure the Kerberos client to associate with a third party server. Valid *kdc_vendor* currently supported are:

  `ms_ad`
  : Microsoft Active Directory

  `mit`
  : MIT KDC server

  `heimdal`
  : Heimdal KDC server

  `shishi`
  : Shishi KDC server

  Knowing the administrative password will be required to associate the client with the server if the `ms_ad` option is specified.

`-c` [*filepath*]
: Specifies the pathname to the `krb5.conf(4)` master file, to be copied over to the local host. The path specified normally points to a master copy on a remote host and brought over to the local host by means of NFS.

-d [*dnsarg*]
Specifies the DNS lookup option to be used and specified in the krb5.conf(4) file. Valid *dnsarg* entries are: none, dns_lookup_kdc, dns_lookup_realm and dns_fallback. Any other entry is considered invalid. The latter three *dnsarg* values assume the same meaning as those described in krb5.conf. dns_lookup_kdc implies DNS lookups for the KDC and the other servers. dns_lookup_realm is for host/domain name-to-realm mapping by means of DNS. dns_fallback is a superset and does DNS lookups for both the servers and the host/domain name-to-realm mapping. A lookup option of none specifies that DNS is not be used for any kind of mapping lookup.

-D *domain_list*
Specifies the host and/or domain names to be mapped to the Kerberos client's default realm name. *domain_list* is a comma–separated list, for example "example.com,host1.example.com". If the -D option is not used, then only the client's domain is used for this mapping. For example, if the client is host1.eng.example.com, then the domain that is mapped to the EXAMPLE.COM realm is example.com.

-K
Configure the Kerberos client without service keys, which are usually stored in /etc/krb5/krb5.keytab. This is useful in the following scenarios:

- The client IP address is dynamically assigned and therefore does not host Kerberized services.
- Client has a static IP address, but does not want to host any Kerberized services.
- Client has a static IP address, but the local administrator does not currently have service keys available for the machine. It is expected that, at a later time, these keys will be installed on the machine.

-f [*fqdn_list*]
This option creates a service principal entry (host/nfs/root) associated with each of the listed fqdn's, if required, and subsequently adds the entries to the local host's keytab.

fqdn_list is a comma-separated list of one or more fully qualified DNS domain names.

This option is especially useful in Kerberos realms having systems offering kerberized services, but situated in multiple different DNS domains.

-h *logical_host_name*
Specifies that the Kerberos client is a node in a cluster. The *logical_host_name* is the logical host name given to the cluster. The resulting /etc/krb5/krb5.conf and /etc/krb5/krb5.keytab files must be manually copied over to the other members of the cluster.

-m *master_kdc*
This option specifies the master KDC to be used by the Kerberos client. *master_kdc* is the host name of the master KDC for the client. If the -m option is not used, then it is assumed that the first KDC host name listed with the -k option is the master KDC.

`-p` [*profile*]
Specifies the profile to be used to enable the reading in of the values of all the parameters required for setup of the machine as a Kerberos client.

The profile should have entries in the format:

PARAM <*value*>

Valid *PARAM* entries are: `REALM`, `KDC`, `ADMIN`, `FILEPATH`, `NFS`, `DNSLOOKUP`, `FQDN`, `NOKEY`, `NOSOL`, `LHN`, `KDCVENDOR`, `RMAP`, `MAS`, and `PAM`.

These profile entries correspond to the `-R` [*realm*], `-k` [*kdc*], `-a` [*adminuser*], `-c` [*filepath*], `-n`, `-d` [*dnsarg*], `-f` [*fqdn_list*], `-K`, `-h` [*logical_host_name*], `-T` [*kdc_vendor*], `-D` [*domain_list*], `-m` [*master_kdc*], and `-s` [*pam_service*] command-line options, respectively. Any other *PARAM* entry is considered invalid and is ignored.

The NFS profile entry can have a value of 0 (do nothing) or 1 (operation is requested). Any other value is considered invalid and is ignored.

Keep in mind that the command line options override the *PARAM* values listed in the profile.

`-s` *pam_service*:*auth_type*[,...]
Specifies that the PAM service names, listed in *pam_service*, are authenticated through Kerberos. Using this option updates `pam.conf(4)` to include a separate authentication stack with `pam_krb5(5)`. Examples of *pam_service* names are `sshd-kbdint`, `xscreensaver`, and so forth.

auth_type can be one of the following keywords:

`first`
Try authenticating through Kerberos first. If this fails try to authenticate through Unix.

`only`
Try to authenticate only through Kerberos.

`optional`
Try authenticating through Unix first. If this is successful try to authenticate through Kerberos.

使用例

例 1 Setting Up a Kerberos Client Using Command-Line Options

To setup a Kerberos client using the `clntconfig/admin` administrative principal for realm `'ABC.COM'`, `kdc` ‘`example1.com`’ and that also does kerberized NFS, enter:

```
# /usr/sbin/kclient -n -R ABC.COM -k example1.com -a clntconfig
```

Alternatively, to set up a Kerberos client using the `clntconfig/admin` administrative principal for the realm ‘`EAST.ABC.COM`’, `kdc` ‘`example2.east.abc.com`’ and that also needs service principal(s) created and/or added to the local `keytab` for multiple DNS domains, enter:

例 1 Setting Up a Kerberos Client Using Command-Line Options (続き)

```
# /usr/sbin/kclient -n -R EAST.ABC.COM -k example2.east.abc.com \
-f west.abc.com,central.abc.com -a clntconfig
```

Note that the krb5 administrative principal used by the administrator needs to have only add, inquire, change-pwd and modify privileges (for the principals in the KDC database) in order for the kclient utility to run. A sample kadm5.acl(4) entry is:

```
clntconfig/admin@ABC.COM acmi
```

例 2 Setting Up a Kerberos Client Using the Profile Option

To setup a Kerberos client using the clntconfig/admin administrative principal for realm 'ABC.COM', kdc 'example1.com' and that also copies over the master krb5.conf from a specified location, enter:

```
# /usr/sbin/kclient -p /net/example1.com/export/profile.krb5
```

The contents of profile.krb5:

```
REALM ABC.COM
KDC example1
ADMIN clntconfig
FILEPATH /net/example1.com/export/krb5.conf
NFS 0
DNSLOOKUP none
```

例 3 Setting Up a Kerberos Client That Has a Dynamic IP Address

In this example a Kerberos client is a DHCP client that has a dynamic IP address. This client does not wish to host any Kerberized services and therefore does not require a keytab (/etc/krb5/krb5.keytab) file.

For this type of client the administrator would issue the following command to configure this machine to be a Kerberos client of the ABC.COM realm with the KDC server kdc1.example.com:

```
# /usr/sbin/kclient -K -R EXAMPLE.COM -k kdc1.example.com
```

ファイル

/etc/krb5/kadm5.acl
: Kerberos access control list (ACL) file.

/etc/krb5/krb5.conf
: Default location for the local host's configuration file.

/etc/krb5/krb5.keytab
: Default location for the local host's keytab file.

/etc/nfssec.conf
: File listing NFS security modes.

/etc/resolv.conf
DNS resolver configuration file.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

encrypt(1), ksh(1), ldapdelete(1), ldapmodify(1), ldapsearch(1), dd(1M), smbadm(1M), kadm5.acl(4), krb5.conf(4), nfssec.conf(4), pam.conf(4), resolv.conf(4), attributes(5), pam_krb5(5)

注意事項

fqdn stands for the Fully Qualified Domain Name of the local host. The kclient utility saves copies of both the krb5.conf(4) and nfssec.conf(4) files to files with corresponding names and .sav extensions. The optional copy of the krb5.conf(4) master file is neither encrypted nor integrity-protected and it takes place over regular NFS.

名前

kdb5_ldap_util – Kerberos configuration utility

形式

```
kdb5_ldap_util  [-D user_dn [-w passwd]] [-H ldap_uri] command
      [command_options]
```

機能説明

The `kdb5_ldap_util` utility allows an administrator to manage realms, Kerberos services, and ticket policies. The utility offers a set of general options, described under OPTIONS, and a set of commands, which, in turn, have their own options. Commands and their options are described in their own subsections, below.

オプション

`kdb5_ldap_util` has a small set of general options that apply to the `kdb5_ldap_util` utility itself and a larger number of options that apply to specific commands. A number of these command-specific options apply to multiple commands and are described in their own section, below.

General Options

The following general options are supported:

`-D` *user_dn*
: Specifies the distinguished name (DN) of a user who has sufficient rights to perform the operation on the LDAP server.

`-H` *ldap_uri*
: Specifies the URI of the LDAP server.

`-w` *passwd*
: Specifies the password of *user_dn*. This option is not recommended.

Common Command-specific Options

The following options apply to a number of `kdb5_ldap_util` commands.

`-subtrees` *subtree_dn_list*
: Specifies the list of subtrees containing the principals of a realm. The list contains the DNs of the subtree objects separated by a colon.

`-sscope` *search_scope*
: Specifies the scope for searching the principals under a subtree. The possible values are 1 or `one` (one level), 2 or `sub` (subtrees).

`-containerref` *container_reference_dn*
: Specifies the DN of the container object in which the principals of a realm will be created. If the container reference is not configured for a realm, the principals will be created in the realm container.

`-maxtktlife` *max_ticket_life*
: Specifies maximum ticket life for principals in this realm.

`-maxrenewlife` *max_renewable_ticket_life*
: Specifies maximum renewable life of tickets for principals in this realm.

`-r` *realm*
: Specifies the Kerberos realm of the database; by default the realm returned by `krb5_default_local_realm(3)` is used.

kdb5_ldap_util Commands

The `kdb5_ldap_util` utility comprises a set of commands, each with its own set of options. These commands are described in the following subsections.

The create Command

The `create` command creates a realm in a directory. The command has the following syntax:

```
create \
[-subtrees subtree_dn_list]
[-sscope search_scope]
[-containerref container_reference_dn]
[-k mkeytype]
[-m|-P password| -sf stashfilename]
[-s]
[-r realm]
[-maxtktlife max_ticket_life]
[-kdcdn kdc_service_list]
[-admindn admin_service_list]
[-maxrenewlife max_renewable_ticket_life]
[ticket_flags]
```

The `create` command has the following options:

`-subtree` *subtree_dn_list*
: See “Common Command-specific Options,” above.

`-sscope` *search_scope*
: See “Common Command-specific Options,” above.

`-containerref` *container_reference_dn*
: See “Common Command-specific Options,” above.

`-k` *mkeytype*
: Specifies the key type of the master key in the database; the default is that given in `kdc.conf(4)`.

`-m`
: Specifies that the master database password should be read from the TTY rather than fetched from a file on the disk.

`-P` *password*
: Specifies the master database password. This option is not recommended.

`-sf` *stashfilename*
: Specifies the stash file of the master database password.

`-s`
: Specifies that the stash file is to be created.

`-maxtktlife` *max_ticket_life*
: See “Common Command-specific Options,” above.

`-maxrenewlife` *max_renewable_ticket_life*
: See “Common Command-specific Options,” above.

`-r` *realm*
: See "Common Command-specific Options," above.

ticket_flags
: Specifies the ticket flags. If this option is not specified, by default, none of the flags are set. This means all the ticket options will be allowed and no restriction will be set. See "Ticket Flags" for a list and descriptions of these flags.

The modify Command

The `modify` command modifies the attributes of a realm. The command has the following syntax:

```
modify \
[-subtrees subtree_dn_list]
[-sscope search_scope]
[-containerref container_reference_dn]
[-r realm]
[-maxtktlife max_ticket_life]
[-maxrenewlife max_renewable_ticket_life]
[ticket_flags]
```

The `modify` command has the following options:

`-subtree` *subtree_dn_list*
: See "Common Command-specific Options," above.

`-sscope` *search_scope*
: See "Common Command-specific Options," above.

`-containerref` *container_reference_dn*
: See "Common Command-specific Options," above.

`-maxtktlife` *max_ticket_life*
: See "Common Command-specific Options," above.

`-maxrenewlife` *max_renewable_ticket_life*
: See "Common Command-specific Options," above.

`-r` *realm*
: See "Common Command-specific Options," above.

ticket_flags
: Specifies the ticket flags. If this option is not specified, by default, none of the flags are set. This means all the ticket options will be allowed and no restriction will be set. See "Ticket Flags" for a list and descriptions of these flags.

The view Command

The `view` command displays the attributes of a realm. The command has the following syntax:

```
view [-r realm]
```

The `view` command has the following option:

`-r` *realm*
See "Common Command-specific Options," above.

The destroy Command

The `destroy` command destroys a realm, including the master key stash file. The command has the following syntax:

```
destroy [-f] [-r realm]
```

The `destroy` command has the following options:

`-f`
If specified, `destroy` does not prompt you for confirmation.

`-r` *realm*
See "Common Command-specific Options," above.

The list Command

The `list` command displays the names of realms. The command has the following syntax:

```
list
```

The `list` command has no options.

The stashsrvpw Command

The `stashsrvpw` command enables you to store the password for service object in a file so that a KDC and Administration server can use it to authenticate to the LDAP server. The command has the following syntax:

```
stashsrvpw [-f filename] servicedn
```

The `stashsrvpw` command has the following option and argument:

`-f` *filename*
Specifies the complete path of the service password file. The default is:

```
/var/krb5/service_passwd
```

servicedn
Specifies the distinguished name (DN) of the service object whose password is to be stored in file.

The create_policy Command

The `create_policy` command creates a ticket policy in a directory. The command has the following syntax:

```
create_policy \
[-r realm]
[-maxtktlife max_ticket_life]
[-maxrenewlife max_renewable_ticket_life]
[ticket_flags]
policy_name
```

The `create_policy` command has the following options:

`-r` *realm*
See "Common Command-specific Options," above.

`-maxtktlife` *max_ticket_life*
: See "Common Command-specific Options," above.

`-maxrenewlife` *max_renewable_ticket_life*
: See "Common Command-specific Options," above.

ticket_flags
: Specifies the ticket flags. If this option is not specified, by default, none of the flags are set. This means all the ticket options will be allowed and no restriction will be set. See "Ticket Flags" for a list and descriptions of these flags.

policy_name
: Specifies the name of the ticket policy.

The modify_policy Command

The `modify_policy` command modifies the attributes of a ticket policy. The command has the following syntax:

```
modify_policy \
[-r realm]
[-maxtktlife max_ticket_life]
[-maxrenewlife max_renewable_ticket_life]
[ticket_flags]
policy_name
```

The `modify_policy` command has the same options and argument as those for the `create_policy` command.

The view_policy Command

The `view_policy` command displays the attributes of a ticket policy. The command has the following syntax:

```
view_policy [-r realm] policy_name
```

The `view_policy` command has the following options:

`-r` *realm*
: See "Common Command-specific Options," above.

policy_name
: Specifies the name of the ticket policy.

The destroy_policy Command

The `destroy_policy` command destroys an existing ticket policy. The command has the following syntax:

```
destroy_policy [-r realm] [-force] policy_name
```

The `destroy_policy` command has the following options:

`-r` *realm*
: See "Common Command-specific Options," above.

`-force`
Forces the deletion of the policy object. If not specified, you will be prompted for confirmation before the policy is deleted. Enter `yes` to confirm the deletion.

policy_name
Specifies the name of the ticket policy.

The list_policy Command

The `list_policy` command lists the ticket policies in the default or a specified realm. The command has the following syntax:

```
list_policy [-r realm]
```

The `list_policy` command has the following option:

`-r` *realm*
See "Common Command-specific Options," above.

Ticket Flags

A number of `kdb5_ldap_util` commands have `ticket_flag` options. These flags are described as follows:

`{-|+}allow_dup_skey`
`-allow_dup_skey` disables user-to-user authentication for principals by prohibiting principals from obtaining a session key for another user. This setting sets the `KRB5_KDB_DISALLOW_DUP_SKEY` flag. `+allow_dup_skey` clears this flag.

`{-|+}allow_forwardable`
`-allow_forwardable` prohibits principals from obtaining forwardable tickets. This setting sets the `KRB5_KDB_DISALLOW_FORWARDABLE` flag. `+allow_forwardable` clears this flag.

`{-|+}allow_postdated`
`-allow_postdated` prohibits principals from obtaining postdated tickets. This setting sets the `KRB5_KDB_DISALLOW_POSTDATED` flag. `+allow_postdated` clears this flag.

`{-|+}allow_proxiable`
`-allow_proxiable` prohibits principals from obtaining proxiable tickets. This setting sets the `KRB5_KDB_DISALLOW_PROXIABLE` flag. `+allow_proxiable` clears this flag.

`{-|+}allow_renewable`
`-allow_renewable` prohibits principals from obtaining renewable tickets. This setting sets the `KRB5_KDB_DISALLOW_RENEWABLE` flag. `+allow_renewable` clears this flag.

`{-|+}allow_svr`
`-allow_svr` prohibits the issuance of service tickets for principals. This setting sets the `KRB5_KDB_DISALLOW_SVR` flag. `+allow_svr` clears this flag.

`{-|+}allow_tgs_req`
`-allow_tgs_req` specifies that a Ticket-Granting Service (TGS) request for a service ticket for principals is not permitted. This option is useless for most purposes. `+allow_tgs_req` clears this flag. The default is `+allow_tgs_req`. In effect, `-allow_tgs_req` sets the `KRB5_KDB_DISALLOW_TGT_BASED` flag on principals in the database.

`{-|+}allow_tix`
: `-allow_tix` forbids the issuance of any tickets for principals. `+allow_tix` clears this flag. The default is `+allow_tix`. In effect, `-allow_tix` sets the `KRB5_KDB_DISALLOW_ALL_TIX` flag on principals in the database.

`{-|+}needchange`
: `+needchange` sets a flag in the attributes field to force a password change; `-needchange` clears that flag. The default is `-needchange`. In effect, `+needchange` sets the `KRB5_KDB_REQUIRES_PWCHANGE` flag on principals in the database.

`{-|+}password_changing_service`
: `+password_changing_service` sets a flag in the attributes field marking a principal as a password-change-service principal (a designation that is most often not useful). `-password_changing_service` clears the flag. That this flag has a long name is intentional. The default is `-password_changing_service`. In effect, `+password_changing_service` sets the `KRB5_KDB_PWCHANGE_SERVICE` flag on principals in the database.

`{-|+}requires_hwauth`
: `+requires_hwauth` requires principals to preauthenticate using a hardware device before being allowed to `kinit(1)`. This setting sets the `KRB5_KDB_REQUIRES_HW_AUTH` flag. `-requires_hwauth` clears this flag.

`{-|+}requires_preauth`
: `+requires_preauth` requires principals to preauthenticate before being allowed to `kinit(1)`. This setting sets the `KRB5_KDB_REQUIRES_PRE_AUTH` flag. `-requires_preauth` clears this flag.

使用例

例 1 Using create

The following is an example of the use of the `create` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
create -subtrees o=org -sscope SUB -r ATHENA.MIT.EDU
Password for "cn=admin,o=org":   password entered
Initializing database for realm 'ATHENA.MIT.EDU'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: master key entered
Re-enter KDC database master key to verify: master key re-enteredjjjjjj
```

例 2 Using modify

The following is an example of the use of the `modify` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
modify +requires_preauth -r ATHENA.MIT.EDU
Password for "cn=admin,o=org":   password entered
Password for "cn=admin,o=org":   password entered
```

例 3 Using view

The following is an example of the use of the `view` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
view -r ATHENA.MIT.EDU
          Password for "cn=admin,o=org":
                              Realm Name: ATHENA.MIT.EDU
                                 Subtree: ou=users,o=org
                                 Subtree: ou=servers,o=org
                             SearchScope: ONE
                     Maximum ticket life: 0 days 01:00:00
                  Maximum renewable life: 0 days 10:00:00
                            Ticket flags: DISALLOW_FORWARDABLE REQUIRES_PWCHANGE
```

例 4 Using destroy

The following is an example of the use of the `destroy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
destroy -r ATHENA.MIT.EDU
Password for "cn=admin,o=org": password entered
Deleting KDC database of 'ATHENA.MIT.EDU', are you sure?
(type 'yes' to confirm)? yes
OK, deleting database of 'ATHENA.MIT.EDU'...
```

例 5 Using list

The following is an example of the use of the `list` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu list
Password for "cn=admin,o=org": password entered
Re-enter Password for "cn=admin,o=org": password re-entered
ATHENA.MIT.EDU
OPENLDAP.MIT.EDU
MEDIA-LAB.MIT.EDU
```

例 6 Using stashsrvpw

The following is an example of the use of the `stashsrvpw` command.

```
# kdb5_ldap_util stashsrvpw -f \
/home/andrew/conf_keyfile cn=service-kdc,o=org
Password for "cn=service-kdc,o=org": password entered
Re-enter password for "cn=service-kdc,o=org": password re-entered
```

例 7 Using create_policy

The following is an example of the use of the `create_policy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
create_policy -r ATHENA.MIT.EDU \
-maxtktlife "1  day" -maxrenewlife "1 week" \
```

例 7 Using create_policy (続き)

```
-allow_postdated +needchange -allow_forwardable tktpolicy
Password for "cn=admin,o=org": password entered
```

例 8 Using modify_policy

The following is an example of the use of the `modify_policy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
modify_policy -r ATHENA.MIT.EDU \
-maxtktlife "60 minutes" -maxrenewlife "10 hours" \
+allow_postdated -requires_preauth tktpolicy
Password for "cn=admin,o=org": password entered
```

例 9 Using view_policy

The following is an example of the use of the `view_policy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
view_policy -r ATHENA.MIT.EDU tktpolicy
Password for "cn=admin,o=org": password entered
            Ticket policy: tktpolicy
      Maximum ticket life: 0 days 01:00:00
   Maximum renewable life: 0 days 10:00:00
             Ticket flags: DISALLOW_FORWARDABLE REQUIRES_PWCHANGE
```

例 10 Using destroy_policy

The following is an example of the use of the `destroy_policy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
destroy_policy -r ATHENA.MIT.EDU tktpolicy
Password for "cn=admin,o=org": password entered
This will delete the policy object 'tktpolicy', are you sure?
(type 'yes' to confirm)? yes
** policy object 'tktpolicy' deleted.
```

例 11 Using list_policy

The following is an example of the use of the `list_policy` command.

```
# kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu \
list_policy -r ATHENA.MIT.EDU
Password for "cn=admin,o=org": password entered
tktpolicy
tmppolicy
userpolicy
```

例 12 Using setsrvpw

The following is an example of the use of the `setsrvpw` command.

例 12 Using setsrvpw (続き)

```
# kdb5_ldap_util setsrvpw -D cn=admin,o=org setsrvpw \
-fileonly -f /home/andrew/conf_keyfile cn=service-kdc,o=org
Password for "cn=admin,o=org": password entered
Password for "cn=service-kdc,o=org": password entered
Re-enter password for "cn=service-kdc,o=org": password re-entered
```

例 13 Using create_service

The following is an example of the use of the `create_service` command.

```
# kdb5_ldap_util -D cn=admin,o=org create_service \
-kdc -randpw -f /home/andrew/conf_keyfile cn=service-kdc,o=org
Password for "cn=admin,o=org": password entered
File does not exist. Creating the file /home/andrew/conf_keyfile...
```

例 14 Using modify_service

The following is an example of the use of the `modify_service` command.

```
# kdb5_ldap_util -D cn=admin,o=org modify_service \
-realm ATHENA.MIT.EDU cn=service-kdc,o=org
Password for "cn=admin,o=org": password entered
Changing rights for the service object. Please wait ... done
```

例 15 Using view_service

The following is an example of the use of the `view_service` command.

```
# kdb5_ldap_util -D cn=admin,o=org view_service \
cn=service-kdc,o=org
Password for "cn=admin,o=org": password entered
                      Service dn: cn=service-kdc,o=org
                    Service type: kdc
                Service host list:
                    Realm DN list: cn=ATHENA.MIT.EDU,cn=Kerberos,cn=Security
```

例 16 Using destroy_service

The following is an example of the use of the `destroy_service` command.

```
# kdb5_ldap_util -D cn=admin,o=org destroy_service \
cn=service-kdc,o=org
Password for "cn=admin,o=org": password entered
This will delete the service object 'cn=service-kdc,o=org', are you sure?
(type 'yes' to confirm)? yes
** service object 'cn=service-kdc,o=org' deleted.
```

例 17 Using list_service

The following is an example of the use of the `list_service` command.

例 17 Using list_service (続き)

```
# kdb5_ldap_util -D cn=admin,o=org list_service
Password for "cn=admin,o=org": password entered
cn=service-kdc,o=org
cn=service-adm,o=org
cn=service-pwd,o=org
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/security/kerberos-5
Interface Stability	Volatile

関連項目

kinit(1), kadmin(1M), kdc.conf(4), attributes(5)

名前

kdb5_util – Kerberos Database maintenance utility

形式

```
/usr/sbin/kdb5_util [-d dbname] [-k mkeytype] [-kv mkeyVNO] [-m ]
      [-M mkeyname] [-P password] [-r realm] [-sf stashfilename]
      [-x db_args]... cmd
```

機能説明

The `kdb5_util` utility enables you to create, dump, load, and destroy the Kerberos V5 database. You can also use `kdb5_util` to create a stash file containing the Kerberos database master key.

オプション

The following options are supported:

`-d` *dbname*
Specify the database name. `.db` is appended to whatever name is specified. You can specify an absolute path. If you do not specify the `-d` option, the default database name is `/var/krb5/principal`.

`-k` *mkeytype*
Specify the master key type. Valid values are `des3-cbc-sha1`, `des-cbc-crc`, `des-cbc-md5`, `des-cbc-raw`, `arcfour-hmac-md5`, `arcfour-hmac-md5-exp`, `aes128-cts-hmac-sha1-96`, and `aes256-cts-hmac-sha1-96`.

`-m`
Enter the master key manually.

`-M` *mkeyname*
Specify the master key name.

`-P` *password*
Use the specified *password* instead of the stash file.

`-r` *realm*
Use *realm* as the default database realm.

`-sf` *stashfile_name*
Specifies the stash file of the master database password.

`-x` *db_args*
Pass database-specific arguments to `kadmin`. Supported arguments are for LDAP and the `Berkeley-db2` plug-in. These arguments are:

`binddn=`*binddn*
LDAP simple bind DN for authorization on the directory server. Overrides the `ldap_kadmind_dn` parameter setting in `krb5.conf(4)`.

`bindpwd=`*bindpwd*
Bind password.

`dbname=`*name*
For the `Berkeley-db2` plug-in, specifies a name for the Kerberos database.

`nconns=`*num*
Maximum number of server connections.

`port=`*num*
Directory server connection port.

オペランド

The following operands are supported:

cmd
Specifies whether to create, destroy, dump, or load the database, or to create a stash file.

You can specify the following commands:

`create -s`
Creates the database specified by the `-d` option. You will be prompted for the database master password. If you specify `-s`, a stash file is created as specified by the `-f` option. If you did not specify `-f`, the default stash file name is `/var/krb5/.k5.realm`. If you use the `-f`, `-k`, or `-M` options when you create a database, then you must use the same options when modifying or destroying the database.

`destroy`
Destroys the database specified by the `-d` option.

`stash`
Creates a stash file. If `-f` was not specified, the default stash file name is `/var/krb5/.k5.realm`. You will be prompted for the master database password. This command is useful when you want to generate the stash file from the password.

`dump` [`-old`] [`-b6`] [`-b7`] [`-ov`] [`-r13`] [`-verbose`] [`-mkey_convert`] [`-new_mkey_file` *mkey_file*] [`-rev`] [`-recurse`] [*filename* [*principals*...]]
Dumps the current Kerberos and KADM5 database into an ASCII file. By default, the database is dumped in current format, "`kdb5_util load_dumpversion 6`". If *filename* is not specified or is the string "-", the dump is sent to standard output. Options are as follows:

`-old`
Causes the dump to be in the Kerberos 5 Beta 5 and earlier dump format ("`kdb5_edit load_dump version 2.0`").

`-b6`
Causes the dump to be in the Kerberos 5 Beta 6 format ("`kdb5_edit load_dump version 3.0`").

`-b7`
Causes the dump to be in the Kerberos 5 Beta 7 format ("`kdb5_util load_dump version 4`"). This was the dump format produced on releases prior to 1.2.2.

`-ov`
Causes the dump to be in `ovsec_adm_export` format.

`-r13`
Causes the dump to be in the Kerberos 5 1.3 format ("`kdb5_util load_dump version 5`"). This was the dump format produced on releases prior to 1.8.

`-verbose`
Causes the name of each principal and policy to be displayed as it is dumped.

`-mkey_convert`
Prompts for a new master key. This new master key will be used to re-encrypt the key data in the dumpfile. The key data in the database will not be changed.

`-new_mkey_file` *mkey_file*
The filename of a stash file. The master key in this stash file will be used to re-encrypt the key data in the dumpfile. The key data in the database will not be changed.

`-rev`
Dumps in reverse order. This might recover principals that do not dump normally, in cases where database corruption has occured.

`-recurse`
Causes the dump to walk the database recursively (`btree` only). This might recover principals that do not dump normally, in cases where database corruption has occurred. In cases of such corruption, this option will probably retrieve more principals than will the `-rev` option.

`load` [`-old`] [`-b6`] [`-b7`] [`-ov`] [`-r13`] [`-hash`] [`-verbose`] [`-update`] *filename dbname* [*admin_dbname*]
Loads a database dump from *filename* into *dbname*. Unless the `-old` or `-b6` option is specified, the format of the dump file is detected automatically and handled appropriately. Unless the `-update` option is specified, `load` creates a new database containing only the principals in the dump file, overwriting the contents of any existing database. The `-old` option requires the database to be in the Kerberos 5 Beta 5 or earlier format ("`kdb5_edit load_dump version 2.0`").

`-b6`
Requires the database to be in the Kerberos 5 Beta 6 format ("`kdb5_edit load_dump version 3.0`").

`-b7`
Requires the database to be in the Kerberos 5 Beta 7 format ("`kdb5_util load_dump version 4`").

`-ov`
Requires the database to be in `ovsec_adm_import` format. Must be used with the `-update` option.

`-hash`
Requires the database to be stored as a hash. If this option is not specified, the database will be stored as a `btree`. This option is not recommended, as databases stored in hash format are known to corrupt data and lose principals.

`-r13`
Causes the dump to be in the Kerberos 5 1.3 format ("`kdb5_util load_dump version 5`"). This was the dump format produced on releases prior to 1.8.

`-verbose`
Causes the name of each principal and policy to be displayed as it is dumped.

`-update`
Records from the dump file are added to or updated in the existing database. Otherwise, a new database is created containing only what is in the dump file and the old one is destroyed upon successful completion.

filename
Required argument that specifies a path to a file containing database dump.

dbname
Required argument that overrides the value specified on the command line or overrides the default.

admin_dbname
Optional argument that is derived from *dbname* if not specified.

`add_mkey` [`-e` *etype*] [`-s`]
Adds a new master key to the K/M (master key) principal. Existing master keys will remain. The `-e` *etype* option allows specification of the `enctype` of the new master key. The `-s` option stashes the new master key in a local stash file which will be created if it does not already exist.

`use_mkey` *mkeyVNO* [*time*]
Sets the activation time of the master key specified by *mkeyVNO*. Once a master key is active (that is, its activation time has been reached) it will then be used to encrypt principal keys either when the principal keys change, are newly created, or when the `update_princ_encryption` command is run. If the *time* argument is provided, that will be the activation time; otherwise the current time is used by default. The format of the optional time argument is that specified in the Time Formats section of the `kadmin(1M)` man page.

`list_mkeys`
List all master keys from most recent to earliest in K/M principal. The output will show the KVNO, `enctype` and `salt` for each `mkey`, similar to `kadmin getprinc` output. An asterisk (`*`) following an `mkey` denotes the currently active master key.

`purge_mkeys` [`-f`] [`-n`] [`-v`]
Delete master keys from the K/M principal that are not used to protect any principals. This command can be used to remove old master keys from a K/M principal once all principal keys are protected by a newer master key.

`-f`
Does not prompt user.

`-n`
Does a dry run, shows master keys that would be purged, but does not actually purge any keys.

`-v`
Verbose output.

`update_princ_encryption` [`-f`] [`-n`] [`-v`] [*princ-pattern*]
Update all principal records (or only those matching the *princ-pattern* glob pattern) to re-encrypt the key data using the active database master key, if they are encrypted using older versions, and give a count at the end of the number of principals updated. If the `-f` option is not given, ask for confirmation before starting to make changes. The `-v` option causes each principal processed (each one matching the pattern) to be listed, and an indication given as to whether it needed updating or not. The `-n` option causes the actions not to be taken, only the normal or verbose status messages displayed; this implies `-f`, since no database changes will be performed and thus there is little reason to seek confirmation.

使用例

例 1 Creating File that Contains Information about Two Principals

The following example creates a file named `slavedata` that contains the information about two principals, `jdb@ACME.COM` and `pak@ACME.COM`.

```
# /usr/krb5/bin/kdb5_util dump -verbose slavedata
jdb@ACME.COM pak@ACME.COM
```

ファイル

`/var/krb5/principal`
Kerberos principal database.

`/var/krb5/principal.kadm5`
Kerberos administrative database. Contains policy information.

`/var/krb5/principal.kadm5.lock`
Lock file for the Kerberos administrative database. This file works backwards from most other lock files (that is, `kadmin` exits with an error if this file does not exist).

`/var/krb5/principal.ulog`
The update log file for incremental propagation.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

`kpasswd(1)`, `gkadmin(1M)`, `kadmin(1M)`, `kadmind(1M)`, `kadmin.local(1M)`, `kdb5_ldap_util(1M)`, `kproplog(1M)`, `kadm5.acl(4)`, `kdc.conf(4)`, `attributes(5)`, `kerberos(5)`

注意事項

The global `-f` is made obsolete with the `-sf` argument for specifying a non-default stash file location. The global `-f` argument might be removed in a future release of the Solaris operating system. Use caution in specifying `-f` as it has different semantics in subcommands as distinguished from its use as a global argument.

名前

kdcmgr – set up a Kerberos Key Distribution Center (KDC)

形式

```
/usr/sbin/kdcmgr [-a admprincipal] [-e enctype]
     [-h] [-p pwfile] [-r realm] subcommand
```

機能説明

Use the `kdcmgr` utility to do the following:

- Configure a master Key Distribution Center (KDC) server.
- Configure a slave KDC. This assumes that a master KDC has already been configured. The default propagation method configured is incremental propagation. See `kpropd(1M)`.
- Specify a list of slave KDCs to configure service principals and create access control list for those slaves on the master KDC.

If you specify no options, `kdcmgr` prompts you for required information, including a password to generate the master key and a password for the administrative principal. When you specify sufficient options, you are still prompted for these passwords, unless you specified the `-p` *pwfile* option.

The `kdcmgr` utility must be run as superuser or by someone who has the Primary Administrator role. The command must be run on the server from which it is invoked.

Note that `kdcmgr` requires the user to enter sensitive information, such as the password used to generate the database's master key and the password for the administrative principal. Great care must be taken to ensure that the connection to the server is secured over the network, by using a protocol such as `ssh(1)`.

You must also exercise great care when selecting the administrative and master key passwords. They should be derived from non-dictionary words and a long string of characters consisting of all of the following character classes:

- special characters (for example, !@#$%^&*)
- numerals (0-9)
- uppercase letters
- lowercase letters

オプション

The following options are supported:

`-a` *admprincipal*

When creating a master KDC, specifies the administrative principal, *admprincipal*, that will be created.

When creating a slave KDC, *admprincipal* is used to authenticate as the administrative principal.

If you omit `-a`, the suggested default administrative principal name is the output of `logname(1)` appended by `/admin`.

`-e` *enctype*
Specifies the encryption type to be used when creating the key for the master key, which is used to encrypt all principal keys in the database. The set of valid encryption types used here are described in `krb5.conf(4)` under the `permitted_enctypes` option. Note that the encryption type specified here must be supported on all KDCs or else they will not be able to decrypt any of the principal keys. Solaris 9 and earlier releases support only the `des-cbc-crc` encryption type for the master key. Therefore, if any of the master or slave KDCs are of these older releases, then `-e des-cbc-crc` would need to be specified on all KDCs configured with `kdcmgr`.

The default encryption type is `aes256-cts-hmac-sha1-96`.

`-h`
Displays usage information for `kdcmgr`.

`-p` *pwfile*
Provides the location of the password file that contains the password used to create the administrative principal and/or master key.

Warning: This option should be used with great care. Make sure that this *pwfile* is accessible only by a privileged user and on a local file system. Once the KDC has been configured, you should remove *pwfile*.

`-r` *realm*
Set the default realm for this server.

If the `-r` option is not specified, `kdcmgr` attempts to obtain the machine's local domain name by submitting the canonical form of the machine's host name to DNS and using the return value to derive the domain name. If successful, the domain name is converted to uppercase and proposed as the default realm name.

Subcommands

The following subcommands are supported:

`create` [*master*]
`create` [`-m` *masterkdc*] slave
Creates a KDC. If no option is specified, an attempt to create a master KDC is made.

`create` [*master*]
Create a master KDC. Upon successful configuration the `krb5kdc(1M)` and `kadmind(1M)` are enabled on the machine.

`create` [`-m` *masterkdc*] slave
Configures a slave KDC. After configuration, the `krb5kdc(1M)` and `kpropd(1M)` services are enabled on the machine.

masterkdc specifies the master KDC to authenticate and with which to perform administrative tasks. If the `-m` option is not specified, you are prompted for a master KDC host name.

`destroy`
Remove all Kerberos configuration and database files associated with the KDC server. A confirmation is required before these files are deleted.

`status`
Determines the role of the KDC, master or slave, and outputs this and the state of such associated processes as:

- `krb5kdc(1M)`
- `kadmind(1M)`
- `kpropd(1M)`

The subcommand also displays information on incremental propagation if the configuration has this feature enabled, as well as any issues with dependent files.

使用例

例 1 Setting up a Master KDC

The following command configures a master KDC with the administrative principal `user1/admin` and with the realm name `EXAMPLE.COM`:

```
$ kdcmgr -a user1/admin -r EXAMPLE.COM create
```

Note that a password will be required to assign to the newly created **`user1/admin`** principal. The password for the master key will also need to be provided.

例 2 Setting up a Slave KDC

The following command configures a slave KDC, authenticates with the administrative principal `user1/admin`, specifies `kdc1` as the master, and uses the `EXAMPLE.COM` realm name:

```
$ kdcmgr -a user1/admin -r EXAMPLE.COM create -m kdc1 slave
```

Note that you must enter the correct password for `user1/admin` and that the master KDC must already have been created before entering this command. The correct password for the master key is also required.

ファイル

`/etc/krb5/krb5.conf`
Main Kerberos configuration file.

`/etc/krb5/kdc.conf`
KDC configuration, used by both master and slave servers.

`/etc/krb5/krb5.keytab`
Default location of the local host's service keys.

`/etc/krb5/kadm5.acl`
Kerberos administrative access control list (ACL).

`/etc/krb5/kadm5.keytab`
Service keys specific to `kadmind(1M)`.

`/var/krb5/principal`
Kerberos principal database.

`/var/krb5/principal.kadm5`
Kerberos policy database.

`/etc/krb5/kpropd.acl`
Used by slaves to indicate from which server to receive updates.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	See below

The command line interface (CLI) is Uncommitted. The CLI output is Not an Interface.

関連項目

`logname(1)`, `ssh(1)`, `kadmin(1M)`, `kadmind(1M)`, `kdb5_util(1M)`, `kdb5_ldap_util(1M)`, `kpropd(1M)`, `krb5kdc(1M)`, `ping(1M)`, `svcadm(1M)`, `kdc.conf(4)`, `krb5.conf(4)`, `attributes(5)`

名前

kernel – UNIX system executable file containing basic operating system services

形式

`kernel-name [-asrvx] [-m smf_options] [-i altinit]`

機能説明

The operating system image, or kernel, is the collection of software comprising the image files (`unix` and `genunix`) and the modules loaded at any instant in time. The system will not function without a kernel to control it.

The kernel is loaded by the `boot(1M)` command in a machine-specific way. The kernel may be loaded from disk, CD-ROM, or DVD (`diskfull boot`) or over the network (`diskless boot`). In either case, the directories under `/platform` and `/kernel` must be readable and must contain executable code which is able to perform the required kernel service. If the `-a` flag is given, the user is able to supply different pathnames for the default locations of the kernel and modules. See `boot(1M)` for more information on loading a specific kernel.

The `moddir` variable contains a list of module directories separated by whitespace. moddir can be set in the `/etc/system` file. The minimal default is:

```
/platform/platform-name/kernel /kernel /usr/kernel
```

This default can be supplemented by a specific platform. It is common for many SPARC systems to override the default path with:

```
/platform/platform-name/kernel:/platform/hardware-class-name\
     /kernel:/kernel:/usr/kernel
```

where *platform-name* can be found using the `-i` option of `uname(1)`, and *hardware-class-name* can be found using the `-m` option of `uname(1)`.

The kernel configuration can be controlled using the `/etc/system` file (see `system(4)`).

`genunix` is the platform-independent component of the base kernel.

オプション

The following options are supported:

`-a`
: Asks the user for configuration information, such as where to find the system file, where to mount `root`, and even override the name of the kernel itself. Default responses will be contained in square brackets ([ ]), and the user may simply enter RETURN to use the default response (note that RETURN is labeled ENTER on some keyboards). To help repair a damaged `/etc/system` file, enter **`/dev/null`** at the prompt that asks for the pathname of the system configuration file. See `system(4)`.

`-i` *altinit*
: Select an alternative executable to be the primordial process. *altinit* must be a valid path to an executable. The default primordial process is `init(1M)`.

`-m` *smf_options*
: The *smf_options* include two categories of options to control booting behavior of the service management facility: recovery options and messages options.

Message options determine the type and amount of messages that smf(5) displays during boot. Service options determine the services which are used to boot the system.

Recovery options

debug
Prints standard per-service output and all `svc.startd` messages to log.

milestone=[milestone]
Boot with some SMF services temporarily disabled, as indicated by *milestone*. *milestone* can be “none”, “single-user”, “multi-user”, “multi-user-server”, or “all”. See the `milestone` subcommand of svcadm(1M).

Messages options

quiet
Prints standard per-service output and error messages requiring administrative intervention.

verbose
Prints standard per-service output with more informational messages.

`-r`
Reconfiguration boot. The system will probe all attached hardware devices and configure the logical namespace in `/dev`. See add_drv(1M) and rem_drv(1M) for additional information about maintaining device drivers.

`-s`
Boots only to init level 's'. See init(1M).

`-v`
Boots with verbose messages enabled. If this flag is not given, the messages are still printed, but the output is directed to the system logfile. See syslogd(1M).

`-x`
Does not boot in clustered mode. This option only has an effect when a version of Sun Cluster software that supports this option has been installed.

使用例

See boot(1M) for examples and instructions on how to boot.

ファイル

`/kernel`
Contains kernel components common to all platforms within a particular instruction set that are needed for booting the system. of the core image file.

`/platform/`*platform-name*`/kernel`
The platform-specific kernel components.

`/platform/`*hardware-class-name*`/kernel`
The kernel components specific to this hardware class.

`/usr/kernel`
: Contains kernel components common to all platforms within a particular instruction set.

The directories in this section can potentially contain the following subdirectories:

drv
: Loadable device drivers

exec
: The modules that execute programs stored in various file formats.

fs
: File system modules

misc
: Miscellaneous system-related modules

sched
: Operating system schedulers

strmod
: System V STREAMS loadable modules

sys
: Loadable system calls

SPARC

cpu
: Processor specific modules

tod
: Time-Of-Day hardware interface modules

As only 64-bit SPARC platforms are supported, all SPARC executable modules are contained within `sparcv9` directories in the directories listed above.

x86

mach
: x86 hardware support

Modules comprising the 32-bit x86 kernel are contained in the above directories, with the 64-bit x86 kernel components contained within `amd64` subdirectories.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/processor, system/library/processorx

関連項目

`kmdb(1)`, `uname(1)`, `isainfo(1)`, `add_drv(1M)`, `boot(1M)`, `init(1M)`, `rem_drv(1M)`, `savecore(1M)`, `svc.startd(1M)`, `svcadm(1M)`, `syslogd(1M)`, `system(4)`, `attributes(5)`, `smf(5)`, `devfs(7FS)`

SPARC Only monitor(1M)

診断

The kernel gives various warnings and error messages. If the kernel detects an unrecoverable fault, it will panic or halt.

注意事項

Reconfiguration boot will, by design, not remove /dev entries for some classes of devices that have been physically removed from the system.

名前

keyserv – server for storing private encryption keys

形式

`keyserv [-c] [-d | -e] [-D] [-n] [-s sizespec]`

機能説明

`keyserv` is a daemon that is used for storing the private encryption keys of each user logged into the system. These encryption keys are used for accessing secure network services such as secure NFS.

Normally, root's key is read from the file `/etc/.rootkey` when the daemon is started. This is useful during power-fail reboots when no one is around to type a password.

`keyserv` does not start up if the system does not have a secure `rpc` domain configured. Set up the domain name by using the `/usr/bin/domainname` command. Usually the `svc:/system/identity:domain` service reads the domain from `/etc/defaultdomain`. Invoking the `domainname` command without arguments tells you if you have a domain set up.

The `/etc/default/keyserv` file contains the following default parameter settings. See ファイル.

`ENABLE_NOBODY_KEYS` Specifies whether default keys for `nobody` are used. `ENABLE_NOBODY_KEYS=NO` is equivalent to the `-d` command-line option. The default value for `ENABLE_NOBODY_KEYS` is `YES`.

オプション

The following options are supported:

`-c` Do not use disk caches. This option overrides any `-s` option.

`-D` Run in debugging mode and log all requests to `keyserv`.

`-d` Disable the use of default keys for `nobody`. See ファイル.

`-e` Enable the use of default keys for `nobody`. This is the default behavior. See ファイル.

`-n` Root's secret key is not read from `/etc/.rootkey`. Instead, `keyserv` prompts the user for the password to decrypt root's key stored in the `publickey` database and then stores the decrypted key in `/etc/.rootkey` for future use. This option is useful if the `/etc/.rootkey` file ever gets out of date or corrupted.

`-s` *sizespec* Specify the size of the extended Diffie-Hellman common key disk caches. The *sizespec* can be one of the following forms:

mechtype`=size` `size` is an integer specifying the maximum number of entries in the cache, or an integer immediately followed by the letter *M*, denoting the maximum size in MB.

`size` This form of *sizespec* applies to all caches.

Note that the `des` mechanism, `AUTH_DES`, does not use a disk cache.

ファイル

`/etc/.rootkey`

`/etc/default/keyserv` Contains default settings. You can use command-line options to override these settings.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`keylogin(1)`, `svcs(1)`, `keylogout(1)`, `svcadm(1M)`, `publickey(4)`, `attributes(5)`, `smf(5)`

注意事項

The `keyserv` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/keyserv:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

killall – kill all active processes

形式

`/usr/sbin/killall [signal]`

機能説明

`killall` is used by `shutdown(1M)` to kill all active processes not directly related to the shutdown procedure.

`killall` terminates all processes with open files so that the mounted file systems will be unbusied and can be unmounted.

`killall` sends *signal* (see `kill(1)`) to the active processes. If no *signal* is specified, a default of 15 is used.

The `killall` command can be run only by the super-user.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`kill(1)`, `ps(1)`, `fuser(1M)`, `shutdown(1M)`, `signal(3C)`, `attributes(5)`

名前

kmscfg – PKCS#11 KMS プロバイダの構成

形式

`kmscfg`

`kmscfg -p[rofile]` *Profile_Name*

`kmscfg -a[gent]` *Agent_ID*

`kmscfg -i[paddr]` *Agent_Address*

`kmscfg -t[imeout]` *Transaction_Timeout*

`kmscfg -f[ailover]` *Failover_Limit*

`kmscfg -d[iscovery]` *Discovery_Freq*

機能説明

`kmscfg` コマンドは、Solaris 暗号化フレームワークで使用できるように PKCS#11 KMS プロバイダ (`pkcs11_kms`) を初期化するために使用します。KMS プロバイダが Oracle Key Manager (OKM) と通信するためには、いくつかの構成情報が使用できるようになっている必要があります。この構成データには、使用されるプロファイルの名前、OKM エージェントの名前、OKM アプライアンス (KMA) の IP アドレス、ほかのいくつかのパラメータ (「形式」を参照) などの情報が含まれています。

デフォルトでは、`kmscfg` はその構成情報を `/var/user/kms/$USERNAME` に格納します。このディレクトリが存在しない場合は、作成されます。構成が検出済みの場合、ユーザーが既存のデータを上書きすることも可能です。デフォルトの場所は、`KMSTOKEN_DIR` 環境変数を使って上書きできます。これは、`kmscfg` を呼び出す前に設定する必要があります。

`kmscfg` を実行する前に、OKM 管理者はアプライアンスそのものに対して必要な初期化および構成手順を実行して、PKCS11 KMS コンシューマが使用する各プロファイルおよびエージェントを設定しておく必要があります。これらのプロファイルの設定手順は、Oracle の Web サイト (`http://docs.oracle.com`) にある『Oracle Key Manager Administration Guide』で入手できます。

管理者による KMA の設定が完了したら、Oracle Solaris クライアントで `kmscfg` を実行してプロバイダを初期化できるように、必要な識別情報 (プロファイル名、エージェント ID、IP アドレス) を提供する必要があります。

オプション

次に示すオプションがサポートされています。コマンド行にプロファイル、エージェント ID、または KMA アドレスを指定しない場合、`kmscfg` によってこれらの項目を入力するように求められます。

`-a` *Agent_ID*
: OKM で設定されたユーザーエージェント ID。KMS トークンの設定に使用されます。プロファイルとエージェント ID が同じであることは珍しいことではありません (例: `MyAgent`)。

`-d` *Discovery_Freq*
: クライアントが OKM クラスタでほかの KMA の使用状況を検出しようとする頻度 (単位は秒)。指定のない場合、*Discovery_Freq* はデフォルトの 10 になります。

-f *Failover_Limit*
クライアントが中止するまでに KMA との通信に失敗できる回数。指定のない場合、*Failover_Limit* はデフォルトの 3 になります。

-i *Agent_Addr*
KMA のアドレス。これは、IPv4 アドレス (*xxx.xxx.xxx.xxx*) でも IPv6 アドレスでもかまいません。クライアントで設定されたネームサービスによって解決できる名前であれば、完全修飾ホスト名も使用できます。OKM クラスタを使用している場合は、クラスタのメンバーのアドレスも指定できます。

-p *Profile_Name*
KMS トークンの設定に使用されるプロファイルの名前。プロファイル名とエージェント ID は同じである必要はありませんが、通常は同じにします。

-t *Transaction_Timeout*
個別の KMS コマンドのタイムアウト期間 (単位は秒)。指定のない場合、この値はデフォルトの 10 になります。

終了ステータス

要求された操作の完了後に、kmscfg は次のいずれかのステータス値を返して終了します。

0
終了に成功しました。

1
失敗。要求された操作を完了できませんでした。

ファイル

/var/kms/$USERNAME
デフォルトの KMS トークン構成ディレクトリ。

${KMSTOKEN_DIR}
代替の KMS トークン構成ディレクトリ。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	/system/library/security/crypto/pkcs11_kms
インタフェースの安定性	流動的

関連項目

pktool(1), attributes(5), pkcs11_kms(5)

『Oracle Key Manager Administration Guide』(http://docs.oracle.com)

注意事項

PKCS#11 クライアントでは、Oracle Key Manager ソフトウェアバージョン 2.4 が OKM にインストールされている必要があります。

PKCS#11 クライアントが複数のシステムから同じエージェント ID を使用する場合は、そのエージェントをワンタイムパスフレーズフラグの設定なしで作成するよう

にしてください。このオプションは、一部のメンバーが 2.4 よりも前のバージョンの OKM ソフトウェアを実行している OKM クラスタでは使用できません。エージェントの作成方法については、『OKM Administration Guide』を参照してください。

PKCS#11 クライアントによる鍵の作成に使用する前に、OKM エージェントにはデフォルトの鍵グループが割り当てられている必要があります。デフォルトの鍵グループがエージェントに割り当てられていない場合、操作は `CKR_PIN_INCORRECT` エラーで失敗します。鍵グループをエージェントに割り当てる方法については、『OKM Administration Guide』を参照してください。

名前

kprop – Kerberos database propagation program

形式

```
/usr/lib/krb5/kprop [-d] [-f file] [-p port-number]
     [-r realm] [-s keytab] [host]
```

機能説明

`kprop` is a command-line utility used for propagating a Kerberos database from a master KDC to a slave KDC. This command must be run on the master KDC. See the Solaris System Administration Guide, Vol. 6 on how to set up periodic propagation between the master KDC and slave KDCs.

To propagate a Kerberos database, the following conditions must be met:

- The slave KDCs must have an `/etc/krb5/kpropd.acl` file that contains the principals for the master KDC and all the slave KDCs.
- A keytab containing a host principal entry must exist on each slave KDC.
- The database to be propagated must be dumped to a file using `kdb5_util(1M)`.

オプション

The following options are supported:

`-d`	Enable debug mode. Default is debug mode disabled.
`-f` *file*	File to be sent to the slave KDC. Default is the `/var/krb5/slave_datatrans` file.
`-p` *port-number*	Propagate *port-number*. Default is port `754`.
`-r` *realm*	Realm where propagation will occur. Default *realm* is the local realm.
`-s` *keytab*	Location of the keytab. Default location is `/etc/krb5/krb5.keytab`.

オペランド

The following operands are supported:

host Name of the slave KDC.

使用例

例 1 Propagating the Kerberos Database

The following example propagates the Kerberos database from the `/tmp/slave_data` file to the slave KDC `london`. The machine `london` must have a host principal keytab entry and the `kpropd.acl` file must contain an entry for the all the KDCs.

```
# kprop -f /tmp/slave_data london
```

ファイル

`/etc/krb5/kpropd.acl`	List of principals of all the KDCs; resides on each slave KDC.
`/etc/krb5/krb5.keytab`	Keytab for Kerberos clients.
`/var/krb5/slave_datatrans`	Kerberos database propagated to the KDC slaves.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5

関連項目

kpasswd(1), svcs(1), gkadmin(1M), inetadm(1M), inetd(1M), kadmind(1M), kadmin.local(1M), kdb5_util(1M), svcadm(1M), kadm5.acl(4), kdc.conf(4), attributes(5), kerberos(5), smf(5)

『Oracle Solaris の管理: セキュリティーサービス』

名前

kpropd – Kerberos propagation daemon for slave KDCs

形式

```
/usr/lib/krb5/kpropd [-d] [-f temp_dbfile] [-F dbfile]
     [-p kdb_util] [-P port_number] [-r realm]
     [-s srv_tabfile] [-S] [-a acl_file]
```

機能説明

The `kpropd` command runs on the slave KDC server. It listens for update requests made by `kprop(1M)` from the master KDC and periodically requests incremental updates from the master KDC.

When the slave receives a kprop request from the master, `kpropd` copies principal data to a temporary text file. Next, `kpropd` invokes `kdb5_util(1M)` (unless a different database utility is selected) to load the text file in database format.

When the slave periodically requests incremental updates, `kpropd` update its `principal.ulog` file with any updates from the master. `kproplog(1M)` can be used to view a summary of the update entry log on the slave KDC.

kpropd is not configured for incremental database propagation by default. These settings can be changed in the `kdc.conf(4)` file:

`sunw_dbprop_enable = [true | false]`	Enables or disables incremental database propagation. Default is `false`.
`sunw_dbprop_slave_poll = N[s, m, h]`	Specifies how often the slave KDC polls for any updates that the master might have. Default is `2m` (two minutes).

The `kiprop/`*<hostname>*`@`*<REALM>* principal must exist in the slave's `keytab` file to enable the master to authenticate incremental propagation requests from the slave. In this syntax, *<hostname>* is the slave KDC's host name and *<REALM>* is the realm in which the slave KDC resides.

オプション

The following options are supported:

`-d`	Enable debug mode. Default is debug mode disabled.
`-f` *temp_dbfile*	The location of the slave's temporary principal database file. Default is `/var/krb5/from_master`.
`-F` *dbfile*	The location of the slave's principal database file. Default is `/var/krb5/principal`.
`-p` *kdb_util*	The location of the Kerberos database utility used for loading principal databases. Default is `/usr/sbin/kdb5_util`.
`-P` *port_number*	Specifies the port number on which `kpropd` will listen. Default is 754 (service name: `krb5_prop`).
`-r` *realm*	Specifies from which Kerberos realm kpropd will receive information. Default is specified in `/etc/krb5/krb5.conf`.

-s *srv_tabfile*	The location of the service table file used to authenticate the kpropd daemon.
-S	Run the daemon in standalone mode, instead of having inetd listen for requests. Default is non-standalone mode.
-a *acl_file*	The location of the kpropd's access control list to verify if this server can run the kpropd daemon. The file contains a list of principal name(s) that will be receiving updates. Default is /etc/krb5/kpropd.acl.

ファイル

/var/krb5/principal	Kerberos principal database.
/var/krb5/principal.ulog	The update log file.
/etc/krb5/kdc.conf	KDC configuration information.
/etc/krb5/kpropd.acl	List of principals of all the KDCs; resides on each slave KDC.
/var/krb5/from_master	Temporary file used by kpropd before loading this to the principal database.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

kdb5_util(1M), kprop(1M), kproplog(1M), kdc.conf(4), krb5.conf(4), attributes(5), kerberos(5)

注意事項

The kprop service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/security/krb5_prop:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

kproplog – display the contents of the Kerberos principal update log

形式

`/usr/sbin/kproplog [-h | -e num]`

機能説明

The `kproplog` displays the contents of the Kerberos principal update log to standard output. This command can be used to keep track of the incremental updates to the principal database, which is enabled by default. The `/var/krb5/principal.ulog` file contains the update log maintained by the `kadmind(1M)` process on the master KDC server and the `kpropd(1M)` process on the slave KDC servers. When updates occur, they are logged to this file. Subsequently any KDC slave configured for incremental updates will request the current data from the master KDC and update their `principal.ulog` file with any updates returned.

The `kproplog` command can only be run on a KDC server by someone with privileges comparable to the superuser. It will display update entries for that server only.

If no options are specified, the summary of the update log is displayed. If invoked on the master, all of the update entries are also displayed. When invoked on a slave KDC server, only a summary of the updates are displayed, which includes the serial number of the last update received and the associated time stamp of the last update.

オプション

The following options are supported:

`-h` Display a summary of the update log. This information includes the database version number, state of the database, the number of updates in the log, the time stamp of the first and last update, and the version number of the first and last update entry.

`-e` *num* Display the last *num* update entries in the log. This is useful when debugging synchronization between KDC servers.

`-v` Display individual attributes per update. If more than one `-v` is specified then very verbose output is displayed. An example of the output generated for one entry:

```
Update Entry
    Update serial # : 4
    Update operation : Add
    Update principal : test@EXAMPLE.COM
    Update size : 424
    Update committed : True
    Update time stamp : Fri Feb 20 23:37:42 2004
    Attributes changed : 6
          Principal
          Key data
          Password last changed
          Modifying principal
          Modification time
          TL data
```

ファイル

`/var/krb5/principal.ulog` The update log file for incremental propagation.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5
Interface Stability	Committed

関連項目

`kpasswd(1)`, `gkadmin(1M)`, `kadmin(1M)`, `kadmind(1M)`, `kdb5_util(1M)`, `kprop(1M)`, `kpropd(1M)`, `kadm5.acl(4)`, `kdc.conf(4)`, `attributes(5)`, `kerberos(5)`

名前

krb5kdc – KDC daemon

形式

```
/usr/lib/krb5/krb5kdc [-d dbpath] [-r realm] [-m]
      [-k masterenctype] [-M masterkeyname]
      [-p port] [-n] [-x db_args]  [-P pid_file]
```

機能説明

krb5kdc is the daemon that runs on the master and slave KDCs to process the Kerberos tickets. For Kerberos to function properly, krb5kdc must be running on at least one KDC that the Kerberos clients can access. Prior to running krb5kdc, you must initialize the Kerberos database using kdb5_util(1M). See the 『Oracle Solaris の管理: セキュリティーサービス』 for information regarding how to set up KDCs and initialize the Kerberos database.

オプション

The following options are supported:

-d *dbpath*
: Specify the path to the database; default value is /var/krb5.

-k *masterenctype*
: Specify the encryption type for encrypting the database. The default value is des-cbc-crc. des3-cbc-sha1, arcfour-hmac-md5, arcfour-hmac-md5-exp, aes128-cts-hmac-sha1-96, and aes256-cts-hmac-sha1-96 are also valid.

-m
: Specify that the master key for the database is to be entered manually.

-M *masterkeyname*
: Specify the principal to retrieve the master Key for the database.

-n
: Specify that krb5kdc should not detach from the terminal.

-p *port*
: Specify the port that will be used by the KDC to listen for incoming requests.

-P *pid_file*
: Tells the KDC to write its PID (followed by a newline) into *pid_file* after it starts up. This can be used to identify whether the KDC is still running and to allow init scripts to stop the correct process.

-r *realm*
: Specify the realm name; default is the local realm name.

-x *db_args*
: Pass database-specific arguments to kadmin. Supported arguments are for the LDAP plug-in. These arguments are:

  binddn=*binddn*
  : Specifies the DN of the object used by the KDC server to bind to the LDAP server. This object should have the rights to read the realm container, principal container and the subtree that is referenced by the realm. Overrides the ldap_kdc_dn parameter setting in krb5.conf(4).

`bindpwd=`*bindpwd*
Specifies the password for the above-mentioned *binddn*. It is recommended not to use this option. Instead, the password can be stashed using the `stashsrvpw` command of `kdb5_ldap_util(1M)`.

`nconns=`*num*
Specifies the number of connections to be maintained per LDAP server.

`host=`*ldapuri*
Specifies, by an LDAP URI, the LDAP server to which to connect.

ファイル

`/var/krb5/principal.db`
Kerberos principal database.

`/var/krb5/principal.kadm5`
Kerberos administrative database. This file contains policy information.

`/var/krb5/principal.kadm5.lock`
Kerberos administrative database lock file. This file works backwards from most other lock files (that is, `kadmin` will exit with an error if this file does *not* exist).

`/etc/krb5/kdc.conf`
KDC configuration file. This file is read at startup.

`/etc/krb5/kpropd.acl`
File that defines the access control list for propagating the Kerberos database using `kprop`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/security/kerberos-5

関連項目

`kill(1)`, `kpasswd(1)`, `gkadmin(1M)`, `kadmind(1M)`, `kadmin.local(1M)`, `kdb5_util(1M)`, `kdb5_ldap_util(1M)`, `logadm(1M)`, `krb5.conf(4)`, `attributes(5)`, `krb5envvar(5)`, `kerberos(5)`,

『Oracle Solaris の管理: セキュリティーサービス』

注意事項

The following signal has the specified effect when sent to the server process using the `kill(1)`command:

`SIGHUP`
`krb5kdc` closes and re-opens log files that it directly opens. This can be useful for external log-rotation utilities such as `logadm(1M)`. If this method is used for log file rotation, set the `krb5.conf(4)` `kdc_rotate` period relation to `never`.

名前

ksslcfg – enable and configure SMF instance of Kernel SSL

形式

```
ksslcfg create -f pkcs11 -T token_label -C certificate_label
     [-d softtoken_directory]
     -p password_file [-u username]
     [-h ca_certchain_file] [-c ciphersuites]
     [-t ssl_session_cache_timeout]
     [-z ssl_session_cache_size] [-v] -x proxy_port [host] ssl_port

ksslcfg create -f pkcs12 -i cert_and_key_pk12file
     -p password_file [-u username]
     [-c ciphersuites] [-t ssl_session_cache_timeout]
     [-z ssl_session_cache_size] [-v] -x proxy_port [host] ssl_port

ksslcfg create -f pem -i cert_and_key_pemfile
     -p password_file [-u username]
     [-c ciphersuites] [-t ssl_session_cache_timeout]
     [-z ssl_session_cache_size] [-v] -x proxy_port [host] ssl_port

ksslcfg delete [-v] [host] ssl_port

ksslcfg -V

ksslcfg -?
```

機能説明

`ksslcfg` manages `smf(5)` instances for the Kernel SSL proxy module. An SSL-enabled web server can use the services of its Kernel SSL proxy to improve the performance of the HTTPS packets processing. It does so by creating an instance of the Kernel SSL service, specifying the SSL proxy port and parameters, and by listening on the proxy port.

The `create` subcommand creates an instance and enables the service for the given address and SSL port.

The `delete` subcommand disables the service for the given address and port, if it is enabled, and deletes the instance from the SMF repository.

`ksslcfg` can be run as root or by other users assigned to the Network Security profile. See `rbac(5)` and `user_attr(4)`.

After `ksslcfg` successfully configures the service in the kernel, the proxy application must be started, or restarted if it is already running.

You must run `ksslcfg` to configure your Kernel SSL proxy before you start your application.

`ksslcfg` allows you to specify an *ssl_port* operand, described under OPERANDS, and, with the `-x` option, a *proxy_port* value. When specified for use with the Kernel SSL proxy, these values cannot also be configured for the Solaris Network Cache and Acceleration (NCA) feature.

The Fault Managed Resource Identifier (FMRI) for the kernel SSL proxy instances is `svc://network/ssl/proxy`. `ksslcfg` creates an instance of that service unique to the

combination of host and SSL port. Instance FMRIs for particular proxy entries can be found with svcs(1) and used for dependencies of other services. The state of the service instance tracks in-kernel configuration only. It does not reflect the presence or state of the application listening on the proxy port.

オプション

The following options are supported:

-c *ciphersuites*
: Set of ciphers a client is allowed to negotiate in a sorted order. The supported SSL version3 and TLS ciphers are listed below. Note that the names are case-insensitive.

```
rsa_rc4_128_sha
rsa_rc4_128_md5
rsa_aes_256_cbc_sha
rsa_aes_128_cbc_sha
rsa_3des_ede_cbc_sha
rsa_des_cbc_sha
```

-f *key_format*
: Uses the certificate/key format specified in *key_format*. The supported options are pkcs11, pkcs12, and pem.

-i *key_and_certificate_file*
: When pkcs12 or pem is specified with the -f option, reads a key and a certificate of the web server from *key_and_certificate_file*. This file can also contain any intermediate CA certificates that form the certificate chain to the root CA for the server certificate. These certificates must follow the server certificate in the file and the order must be bottom up: lowest level CA certificate followed by the next higher level CA certificate, and so on.

-C *certificate_label*
: PKCS#11 can store multiple certificates in single token. This option enables you to specify a single certificate, identified by *certificate_label*. This label must match the CKA_LABEL on the certificate object in the token specified by -T. This option is to be used only with -f pkcs11.

-d *softtoken_directory*
: This option is applicable only with the pkcs11 key format, when the token label is the Sun Software PKCS#11 softtoken. Use this option to override the default location of the PKCS#11 softtoken directory ($HOME/.sunw). See pkcs11_softtoken(5).

-h *ca_certchain_file*
: When pkcs11 is specified with the -f option, reads a set of intermediate CA certificates that form the certificate chain to the root CA for the server certificate (specified with the -C option), from *ca_certchain_file*. The file must be in PEM format.

-p *password_file*
: Obtains the password used to encrypt the private key from *password_file*. When using the pkcs11 option (see -f, above), the password is used to authenticate the user to the PKCS #11 token.

`-t` *ssl_session_cache_timeout*
The timeout value, in seconds, for an SSL session. It corresponds to `SSL3SessionTimeout` of the Sun ONE web server configuration or `SSLSessionCacheTimeout` of `mod_ssl`.

`-T` *token_label*
When `pkcs11` is specified with `-f`, uses the PKCS#11 token specified in *token_label*. Use `cryptoadm list -v` to display all PKCS#11 tokens available.

`-u` *username*
The username of the user who owns the password file. If omitted, the system will try to read the password file as root.

`-v`
Verbose mode.

`-V`
Displays the version.

`-x` *proxy_port*
The SSL proxy port. The port number is designated exclusively for clear-text HTTP communication between the web server and the kernel SSL proxy module. No external HTTP packets are delivered to this port.

`-z` *ssl_session_cache_size*
The maximum number of SSL sessions that can be cached. It corresponds to `SSLCacheEntries` of the Sun ONE web server configuration. When this option is not specified, the default is 5000 entries.

`-?`
Displays the usage of the command.

オペランド

[*host*] [*ssl_port*]
The address and the port of the web server for which the kernel SSL entry is created. If *host* is omitted, the entry will be used for all requests that arrived at the *ssl_port*, regardless of the destination address. Both a host name and an IP address are acceptable forms for *host*. *ssl_port* is required. Typically, this has a value of 443.

使用例

例 1　Create and Enable a Kernel SSL Instance

The following command creates and enables a Kernel SSL instance using a certificate and a key in PKCS#11 format.

```
# ksslcfg create -f pkcs11 -T "Sun Software PKCS#11 softtoken"  \
-C "Server-Cert" -p /some/directory/password -u webservd \
-x 8080 www.mysite.com 443

% svcs svc:/network/ssl/proxy
STATE          STIME    FMRI
online         Sep_27   svc:/network/ssl/proxy:kssl-www-mysite-com-443
```

例 2 Create and Enable a Default Instance for All Addresses

The following command creates and enables a default instance for all addresses from a certificate and key in a pkcs#12 file.

```
# ksslcfg create -x 8888 -f pkcs12 -i /some/directory/keypair.p12 \
    -p /some/directory/password -u webservd 443
```

例 3 Create and Enable an Instance with Specific Cipher Suites

The following command creates and enables an instance with specific cipher suites.

```
# ksslcfg create -x 8080 -f pem \
-i /some/directory/keypair.pem -p /some/directory/password \
-c "rsa_rc4_128_md5,rsa_rc4_128_sha" \
209.249.116.195 443
```

例 4 Disable and Delete an Instance

The following command disables and deletes an instance.

```
# ksslcfg delete www.mysite.com 443
```

例 5 Establishing Dependency of Proxy Application

The sequence of commands shown below establishes a dependency of a proxy application on a KSSL instance. Note that he proxy application should only be started after the SSL kernel proxy instance has been started.

The following commands establish the dependency of an Apache 2.2 web server. KSSL has been configured to listen on SSL port 443 and a wildcard address.

```
# svccfg -s svc:/network/http:apache22
svc:/network/http:apache22> addpg kssl dependency
svc:/network/http:apache22> setprop kssl/entities = fmri:svc:/network/\
ssl/proxy:kssl-INADDR_ANY-443
svc:/network/http:apache22> setprop kssl/grouping = astring: require_all
svc:/network/http:apache22> setprop kssl/restart_on = astring: refresh
svc:/network/http:apache22> setprop kssl/type = astring: service
svc:/network/http:apache22> end
```

Following these commands, enable the web server:

```
# svcadm enable svc:/network/http:apache22
```

If the web server was already running, restart it:

```
# svcadm refresh svc:/network/http:apache22
# svcadm restart svc:/network/http:apache22
```

終了ステータス

`0`
Successful completion.

`>0`
An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

Command line options and the utility name are Committed. The command output, the FMRI service name (`svc://network/ssl/proxy`), and the FMRI instance's name format are Uncommitted

関連項目

`svcprop(1)`, `svcs(1)`, `cryptoadm(1M)`, `svcadm(1M)`, `svccfg(1M)`, `user_attr(4)`, `attributes(5)`, `kssl(5)`, `pkcs11_softtoken(5)`, `rbac(5)`, `smf(5)`, `nca(7d)`

Chown, P., Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), RFC 3268, June 2002.

注意事項

`ksslcfg create` without an host argument creates an `INADDR_ANY` `smf` instance. `ksslcfg delete` without an host argument deletes only the `INADDR_ANY` instance. `ksslcfg delete` needs a host argument to delete any non-`INADDR_ANY` instance.

On a system with `zones(5)` installed, the `ksslcfg` command can be used only in the global zone at this time.

名前

kstat – display kernel statistics

形式

```
kstat [-lpq] [-T u | d ] [-c class] [-m module]
     [-i instance] [-n name] [-s statistic]
     [interval [count]]
```

```
kstat [-lpq] [-T u | d ] [-c class]
     [module:instance:name:statistic]...
     [interval [count]]
```

機能説明

The `kstat` utility examines the available kernel statistics, or kstats, on the system and reports those statistics which match the criteria specified on the command line. Each matching statistic is printed with its module, instance, and name fields, as well as its actual value.

Kernel statistics may be published by various kernel subsystems, such as drivers or loadable modules; each kstat has a module field that denotes its publisher. Since each module might have countable entities (such as multiple disks associated with the `sd`(7D) driver) for which it wishes to report statistics, the kstat also has an instance field to index the statistics for each entity; kstat instances are numbered starting from zero. Finally, the kstat is given a name unique within its module.

Each kstat may be a special kstat type, an array of name-value pairs, or raw data. In the name-value case, each reported value is given a label, which we refer to as the statistic. Known raw and special kstats are given statistic labels for each of their values by `kstat`; thus, all published values can be referenced as *module*:*instance*:*name*:*statistic*.

When invoked without any module operands or options, kstat will match all defined statistics on the system. Example invocations are provided below. All times are displayed as fractional seconds since system boot.

オプション

The tests specified by the following options are logically ANDed, and all matching kstats will be selected. A regular expression containing shell metacharacters must be protected from the shell by enclosing it with the appropriate quotes.

The argument for the `-c`, `-i`, `-m`, `-n`, and `-s` options may be specified as a shell glob pattern, or a Perl regular expression enclosed in '/' characters.

`-c` *class*	Displays only kstats that match the specified class. *class* is a kernel-defined string which classifies the “type” of the kstat.
`-i` *instance*	Displays only kstats that match the specified instance.
`-l`	Lists matching kstat names without displaying values.
`-m` *module*	Displays only kstats that match the specified module.
`-n` *name*	Displays only kstats that match the specified name.
`-p`	Displays output in parseable format. All example output in this document is given in this format. If this option is not specified, `kstat` produces output in a human-readable, table format.

`-q`	Displays no output, but return appropriate exit status for matches against given criteria.
`-s` *statistic*	Displays only kstats that match the specified statistic.
`-T d	u`	Displays a time stamp before each statistics block, either in `date(1)` format (`d`) or as an alphanumeric representation of the value returned by `time(2)` (`u`).

オペランド

The following operands are supported:

module:*instance*:*name*:*statistic*	Alternate method of specifying module, instance, name, and statistic as described above. Each of the module, instance, name, or statistic specifiers may be a shell glob pattern or a Perl regular expression enclosed by '/' characters. It is possible to use both specifier types within a single operand. Leaving a specifier empty is equivalent to using the '*' glob pattern for that specifier.
interval	The number of seconds between reports.
count	The number of reports to be printed.

使用例

In the following examples, all the command lines in a block produce the same output, as shown immediately below. The exact statistics and values will of course vary from machine to machine.

例 1 Using the kstat Command

```
example$ kstat -p -m unix -i 0 -n system_misc -s 'avenrun*'
example$ kstat -p -s 'avenrun*'
example$ kstat -p 'unix:0:system_misc:avenrun*'
example$ kstat -p ':::avenrun*'
example$ kstat -p ':::/^avenrun_\d+min$/'

unix:0:system_misc:avenrun_15min        3
unix:0:system_misc:avenrun_1min 4
unix:0:system_misc:avenrun_5min 2
```

例 2 Using the kstat Command

```
example$ kstat -p -m cpu_stat -s 'intr*'
example$ kstat -p cpu_stat:::/^intr/

cpu_stat:0:cpu_stat0:intr       29682330
cpu_stat:0:cpu_stat0:intrblk    87
cpu_stat:0:cpu_stat0:intrthread 15054222
cpu_stat:1:cpu_stat1:intr       426073
cpu_stat:1:cpu_stat1:intrblk    51
cpu_stat:1:cpu_stat1:intrthread 289668
```

例 2 Using the kstat Command (続き)

```
cpu_stat:2:cpu_stat2:intr       134160
cpu_stat:2:cpu_stat2:intrblk    0
cpu_stat:2:cpu_stat2:intrthread 131
cpu_stat:3:cpu_stat3:intr       196566
cpu_stat:3:cpu_stat3:intrblk    30
cpu_stat:3:cpu_stat3:intrthread 59626
```

例 3 Using the kstat Command

```
example$ kstat -p :::state ':::avenrun*'
example$ kstat -p :::state :::/^avenrun/

cpu_info:0:cpu_info0:state      on-line
cpu_info:1:cpu_info1:state      on-line
cpu_info:2:cpu_info2:state      on-line
cpu_info:3:cpu_info3:state      on-line
unix:0:system_misc:avenrun_15min        4
unix:0:system_misc:avenrun_1min 10
unix:0:system_misc:avenrun_5min 3
```

例 4 Using the kstat Command

```
example$ kstat -p 'unix:0:system_misc:avenrun*' 1 3
unix:0:system_misc:avenrun_15min        15
unix:0:system_misc:avenrun_1min 11
unix:0:system_misc:avenrun_5min 21

unix:0:system_misc:avenrun_15min        15
unix:0:system_misc:avenrun_1min 11
unix:0:system_misc:avenrun_5min 21

unix:0:system_misc:avenrun_15min        15
unix:0:system_misc:avenrun_1min 11
unix:0:system_misc:avenrun_5min 21
```

例 5 Using the kstat Command

```
example$ kstat -p -T d 'unix:0:system_misc:avenrun*' 5 2
Thu Jul 22 19:39:50 1999
unix:0:system_misc:avenrun_15min        12
unix:0:system_misc:avenrun_1min 0
unix:0:system_misc:avenrun_5min 11

Thu Jul 22 19:39:55 1999
unix:0:system_misc:avenrun_15min        12
unix:0:system_misc:avenrun_1min 0
unix:0:system_misc:avenrun_5min 11
```

例 6 Using the kstat Command

```
example$ kstat -p -T u 'unix:0:system_misc:avenrun*'
932668656
unix:0:system_misc:avenrun_15min        14
unix:0:system_misc:avenrun_1min 5
unix:0:system_misc:avenrun_5min 18
```

終了ステータス

The following exit values are returned:

0 One or more statistics were matched.

1 No statistics were matched.

2 Invalid command line options were specified.

3 A fatal error occurred.

ファイル

/dev/kstat kernel statistics driver

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

date(1), sh(1), time(2), gmatch(3GEN), kstat(3KSTAT), attributes(5), kstat(7D), sd(7D), kstat(9S)

注意事項

If the pattern argument contains glob or Perl RE metacharacters which are also shell metacharacters, it will be necessary to enclose the pattern with appropriate shell quotes.

名前

ktkt_warnd – Kerberos warning daemon

形式

/usr/lib/krb5/ktkt_warnd

機能説明

ktkt_warnd is a daemon on Kerberos clients that can warn users when their Kerberos tickets are about to expire or renew the tickets before they expire. It is invoked by inetd when a ticket-granting ticket (TGT) is obtained for the first time, such as after using the kinit command. ktkt_warnd can be configured through the /etc/krb5/warn.conf file on the client. In warn.conf, you can specify that you be supplied notice, through syslog or terminal or mail, of ticket expiration or to renew the TGT.

ファイル

/etc/krb5/warn.conf　Kerberos warning configuration file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

svcs(1), inetadm(1M), inetd(1M), svcadm(1M), warn.conf(4), attributes(5), kerberos(5), smf(5)

注意事項

The ktkt_warnd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/security/ktkt_warn:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

labeld – Trusted Extensions ラベルデーモン

形式

```
/usr/lib/labeld
```

機能説明

label デーモンは、Trusted Extensions が有効な場合に、ブート時に起動するシステムデーモンです。このデーモンは自動的に起動されるので、直接起動しないようにしてください。このデーモンは、プログラミングインタフェースの構成要素ではありません。

Trusted Extensions を有効にするには、次の内容をコマンド行に入力します。

```
% svcadm enable svc:/system/labeld
```

Trusted Extensions はゾーンの初期化および操作に影響を及ぼすため、Trusted Extensions を有効にする前にすべてのゾーンを削除する必要があります。有効にしたあと、Trusted Extensions が適切に初期化されるようにシステムをリブートしてください。

Trusted Extensions の有効化や無効化を行えるのは、solaris.smf.manage.labels 承認を持つユーザーまたは役割だけです(例: Information Security または Object Label Management 権利プロファイルを保持するユーザーや役割、または スーパーユーザー)。

Trusted Extensions の機能を使用する前に、ほかの構成手順を実行する必要があります。詳細は、『Solaris Trusted Extensions インストールと構成』ガイドを参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	不確実

関連項目

svcs(1), svcadm(1M), syslogd(1M), attributes(5), smf(5)

Solaris Trusted Extensions 管理者の手順

注意事項

labels(5) サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/labeld:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

labelit – list or provide labels for file systems

形式

```
labelit [-F FSType] [-V] [-o FSType-specific-options] special
      [operands]
```

機能説明

The `labelit` utility is used to write or display labels on unmounted disk file systems.

オプション

The following options are supported:

`-F` *FSType*
: Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by matching the *special* with an entry in the table. If no matching entry is found, the default file system type specified in `/etc/default/fs` will be used.

`-V`
: Echo complete command line. This option may be used to verify and validate the command line. Additional information obtained using a `/etc/vfstab` lookup is included in the output. The command is not executed.

`-o`
: Specify *FSType*-specific options. See the manual page for the `labelit` module specific to the file system type.

オペランド

The following operands are supported. If no operands are specified, `labelit` will display the value of the labels.

special
: The disk partition (for example, `/dev/rdsk/c0t3d0s6`). The device may not be on a remote machine.

operands
: *FSType*-specific operands. Consult the manual page of the *FSType*-specific `labelit` command for detailed descriptions.

使用法

See `largefile(5)` for the description of the behavior of `labelit` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

終了ステータス

The following exit values are returned:

`0`
: Write or display of labels was successful.

non-zero
: An error occurred.

ファイル

`/etc/vfstab`
: List of default parameters for each file system.

`/etc/default/fs`
: Default local file system type. Default values can be set for the following flags in `/etc/default/fs`. For example:

```
LOCAL=ufs
```

`LOCAL`
: The default partition for a command if no *FSType* is specified.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

labelit_hsfs(1M), labelit_udfs(1M), labelit_ufs(1M), volcopy(1M), vfstab(4), attributes(5), largefile(5)

注意事項

This utility may not be supported for all *FSTypes*.

名前

labelit_hsfs – provide and print labels for hsfs file systems

形式

```
/usr/sbin/labelit -F hsfs [generic_options]
      [-o specific_options] special
```

機能説明

`labelit` can be used to provide labels for unmounted CD-ROM images (CD-ROMs may not be labeled, as they are read-only media).

generic_options are options supported by the generic `labelit` command.

If no *specific_options* are specified, `labelit` prints the current value of all label fields.

The *special* name should be the physical disk section (for example, `/dev/dsk/c0d0s6`).

オプション

`-o` Use one or more of the following *name=value* pairs separated by commas (with no intervening spaces) to specify values for specific label fields. According to the ISO 9660 specification, only certain sets of characters may be used to fill in these labels. Thus, ‘ ‘d-characters" below refers to the characters ‘A' through ‘Z', the digits ‘0' through ‘9', and the ‘_' (underscore) character. “a-characters" below refers to ‘A' through ‘Z', ‘0' through ‘9', space, and the following characters: !"%&'()*+,-./:;<=>?_.

- `absfile=` Abstract file identifier, d-characters, 37 characters maximum.
- `applid=` Application identifier, d-characters, 128 characters maximum.
- `bibfile=` Bibliographic file identifier, d-characters, 37 characters maximum.
- `copyfile=` Copyright file identifier, d-characters, 128 maximum.
- `prepid=` Data preparer identifier, d-characters, 128 maximum.
- `pubid=` Publisher identifier, d-characters, 128 maximum.
- `sysid=` System identifier, a-characters, 32 maximum.
- `volid=` Volume identifier, d-characters, 32 maximum.
- `volsetid=` Volume set identifier, d-characters, 128 maximum.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`labelit(1M)`, `volcopy(1M)`, `attributes(5)`

名前

labelit_udfs – provide and print labels for udf file systems

形式

```
labelit -F  udfs [generic_options] [-o specific_options] special
      [fsname  volume]
```

機能説明

The `labelit` command writes labels on an unmounted disk that contains a universal disk file (udf) system. These labels can be used to identify volumes.

オプション

The following options are supported:

generic_options
: Specify *generic_options* supported by the generic `labelit` command. See `labelit(1M)` for descriptions of supported options.

`-o` *specific_options*
: Specify udfs-file-system-specific options in a comma-separated list with no intervening spaces. The following *specific_options* are available:

  `lvinfo1=`*string*
  : Specify information to be inserted in the `LVInfo1` field of the Implementation Use Volume Descriptor. Information in `LVInfo1` is generally used to identify the person creating the file system. The maximum length of the string specified is 35 bytes.

  `lvinfo2=`*string*
  : Specify information to be inserted into the `LVInfo2` field of the Implementation Use Volume Descriptor. Information in `LVInfo2` is generally used to identify the organization responsible for creating the file system. The maximum length of the string specified is 35 bytes.

  `lvinfo3=`*string*
  : Specify information to be inserted into the `LVInfo3` field of the Implementation Use Volume Descriptor. Information in `LVInfo3` is generally used to identify the contact information for the medium. The maximum length of the string specified is 35 bytes.

オペランド

The following operands are supported:

special
: Specify *special* as the physical disk slice, for example, `/dev/rdsk/c0t0d0s6`. The device can not be on a remote machine.

fsname
: Specify *fsname* as the mount point, (for example, `root`, `u1`, and so forth), of the file system.

volume
: Specify *volume* as the physical volume name.

If none of the options (*fsname*, *volume*, *specific_options*) is specified, `labelit` prints the current values of *fsname*, *volume*, `LVInfo1`, `LVInfo2` and `LVInfo3`.

終了ステータス

The following exit values are returned:

0 Successful completion.

non-zero An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/udfs

関連項目

labelit(1M), attributes(5)

名前

labelit_ufs – provide and print labels for ufs file systems

形式

`labelit -F ufs [generic_options] special [fsname volume]`

機能説明

`labelit` is used to write labels on unmounted disk file systems. Such labels may be used to uniquely identify volumes and are used by volume-oriented programs such as `volcopy(1M)`.

オプション

The following option is supported:

generic_options options supported by the generic `labelit` command. See `labelit(1M)`.

オペランド

The following operands are supported:

special name should be the physical disk section (for example, `/dev/dsk/c0d0s6`). The device may not be on a remote machine.

fsname represents the mount point (for example, `root`, `u1`, and so on) of the file system.

volume may be used to represent the physical volume name.

If *fsname* and *volume* are not specified, `labelit` prints the current values of these labels. Both *fsname* and *volume* are limited to six or fewer characters.

終了ステータス

The following exit values are returned:

`0` Write or display of labels was successful.

non-zero An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`labelit(1M)`, `volcopy(1M)`, `attributes(5)`, `ufs(7FS)`

名前

latencytop – システムおよびアプリケーションの遅延関連の統計を報告する

形式

```
latencytop [-t interval] [-o log_file] [-k log_level]
     [-f [no]feature,...] [-l log_interval] [-h]
     [ -s pid=PID | pgid=PGID ]
```

機能説明

LatencyTOP は、システムおよびアプリケーションの遅延に関する統計を報告する監視ツールです。このツールは、システムまたはシステムで実行中のアプリケーションで発生している遅延の場所および種類に関する統計を報告します。この統計を使用して特定された遅延を取り除くことで、アプリケーションやシステムのパフォーマンススループットを改善できます。

このツールは、システムアクティビティーを定期的に分析して、データを出力ウィンドウに表示します。追跡対象の遅延には、LWP によるスリープの開始と終了、および LWP による順序のスピン (同期オブジェクトを取得するため) の 2 種類があります。このツールは、Solaris DTrace フレームワークを使用して、システムやアプリケーション LWP に関するこれら 2 つの非アクティブ化シナリオに対応する統計を収集します。

出力ウィンドウは 2 つのセクションに分かれています。上部にはシステム全体の統計が、下部には各プロセスの統計がそれぞれ表示されます。ユーザーは (左右の矢印キーを使用して) プロセスリスト内を移動し、必要なリストを選択できます。選択したプロセスの統計は、ウィンドウの下部に表示されます。t または T キーを押すと、選択したプロセスが LWP 固有の表示になります。t または T キーを使って、プロセス表示とスレッド表示を切り替えることができます。

実行中に r または R キーを押すと、分析が強制的に更新されます。時間間隔は、デフォルトまたは指定した値 (-t を使用した場合) に復元されます。アプリケーションを終了するには、q または Q キーを押す必要があります。

オプション

次のオプションがサポートされています。

`-f`、`--feature` [no]*feature1*,[no]*feature2*,...

LatencyTOP の機能を有効または無効にします。機能は、次のいずれかに設定できます。

`[no]filter`

大規模で割り込み可能な遅延 (スリープなど) をフィルタ処理します。デフォルトは `off` です。

`[no]sched`

`sched` (PID=0) を監視します。デフォルトは `off` です。

`[no]sobj`

同期オブジェクトを監視します。デフォルトは `on` です。

`[no]low`

小規模な遅延をサンプリングしてオーバーヘッドを引き下げます。この機能を有効にすると、小規模の遅延が統計的に見積もられるため、CPU 使用率が

低下します。これは、ビジー状態の Web サーバーなど、作業負荷が高い場合に使用します。デフォルトは `off` です。

`-h`
コマンドの使用方法を表示します。

`-k` *log_level*
ログファイルのログ記録レベルを指定します。有効な値は次のとおりです。

`0` `none` (デフォルト)

`1` `unknown`

`2` `all`

`-l` [*log_interval*]
データを *log_interval* 秒ごとにファイルに書き込みます。*log_interval* は 60 より大きい値にする必要があります。

`-o` *log_file*
出力を書き込むログファイルを指定します。デフォルトのログファイルは `/var/log/latencytop.log` です。

`-s` pid=*PID* | pgid=*PGID*
指定したプロセスまたは指定したプロセスグループのみを追跡して、そのプロセスまたはプロセスグループに関連するデータのみを表示します。

`-t` *interval*
ツールがシステムから統計を収集する間隔を秒単位で指定します。指定可能な値は、1 - 60 です。デフォルトは 5 秒です。

使用例

例 1 ツールを実行する

次のコマンドは、デフォルトのオプション値を使用してツールを起動します。

```
% latencytop
```

例 2 間隔を設定する

次のコマンドは、サンプリング間隔を 2 秒に設定します。

```
% latencytop -t 2
```

例 3 ログファイルを指定する

次のコマンドは、ログファイルを `/tmp/latencytop.log` に設定します。

```
% latencytop -o /tmp/latencytop.log
```

例 4 ログレベルを指定する

次のコマンドは、ログレベルを `all` に設定します。

```
% latencytop -l 2
```

例 5 遅延の追跡を有効にする

次のコマンドは、同期オブジェクトにより発生した遅延の追跡を有効にします。

```
% latencytop -f sobj
```

例 6 プロセスグループのデータを表示する

次のコマンドは、プロセスグループ 630 に属するプロセスの追跡データを表示します。

```
% latencytop -s pgid=630
```

終了ステータス

0
正常終了。

1
エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	x86、SPARC
使用条件	diagnostic/latencytop
インタフェースの安定性	流動的

関連項目

dtrace(1M), kstat(1M), attributes(5)

使用法

LatencyTOP を実行するには DTrace 特権が必要です。

名前

ldapaddent – create LDAP entries from corresponding /etc files

形式

```
ldapaddent [-cpv] [-a authenticationMethod] [-b baseDN]
     -D bindDN [-w bind_password] [-j passwdFile] [-f filename]
     database

ldapaddent [-cpv] -a sasl/GSSAPI [-b baseDN] [-f filename]
     database

ldapaddent -d [-v] [-a authenticationMethod] [-D bindDN]
     [-w bind_password] [-j passwdFile] database

ldapaddent [-cpv] -h LDAP_server[:serverPort] [-M domainName]
     [-N profileName]  [-P certifPath] [-a authenticationMethod]
     [-b baseDN] -D bindDN [-w bind_password] [-f filename]
     [-j passwdFile] database

ldapaddent [-cpv] -h LDAP_server[:serverPort] [-M domainName]
     [-N profileName]  [-P certifPath] [-a authenticationMethod]
     [-b baseDN] [-f filename] database

ldapaddent -d [-v] -h LDAP_server[:serverPort] [-M domainName]
     [-N profileName]  [-P certifPath] [-a authenticationMethod]
     [-b baseDN] -D bindDN [-w bind_password] [-j passwdFile]
     database
```

機能説明

`ldapaddent` creates entries in LDAP containers from their corresponding `/etc` files. This operation is customized for each of the standard containers that are used in the administration of Solaris systems. The *database* argument specifies the type of the data being processed. Legal values for this type are one of `aliases`, `auto_*`, `bootparams`, `ethers`, `group`, `hosts` (including both IPv4 and IPv6 addresses), `ipnodes` (alias for `hosts`), `netgroup`, `netmasks`, `networks`, `passwd`, `shadow`, `protocols`, `publickey`, `rpc`, and `services`. In addition to the preceding, the *database* argument can be one of the RBAC-related files (see `rbac(5)`):

- `/etc/user_attr`
- `/etc/security/auth_attr`
- `/etc/security/prof_attr`
- `/etc/security/exec_attr`

By default, `ldapaddent` reads from the standard input and adds this data to the LDAP container associated with the database specified on the command line. An input file from which data can be read is specified using the `-f` option.

If you specify the `-h` option, `ldapaddent` establishes a connection to the server indicated by the option in order to obtain a `DUAProfile` specified by the `-N` option. The entries will be stored in the directory described by the configuration obtained.

By default (if the `-h` option is not specified), entries will be stored in the directory based on the client's configuration. To use the utility in the default mode, the Solaris LDAP client must be set up in advance.

The location where entries are to be written can be overridden by using the `-b` option.

If the entry to be added exists in the directory, the command displays an error and exits, unless the `-c` option is used.

Although, there is a `shadow` database type, there is no corresponding `shadow` container. Both the `shadow` and the `passwd` data is stored in the `people` container itself. Similarly, data from `networks` and `netmasks` databases are stored in the `networks` container.

The `user_attr` data is stored by default in the `people` container. The `prof_attr` and `exec_attr` data is stored by default in the `SolarisProfAttr` container.

You must add entries from the `passwd` database before you attempt to add entries from the `shadow` database. The addition of a `shadow` entry that does not have a corresponding `passwd` entry will fail.

The `passwd` database must precede the `user_attr` database.

For better performance, the recommended order in which the databases should be loaded is as follows:

- `passwd` database followed by `shadow` database
- `networks` database followed by `netmasks` database
- `bootparams` database followed by `ethers` database

Only the first entry of a given type that is encountered will be added to the LDAP server. The `ldapaddent` command skips any duplicate entries.

オプション

The `ldapaddent` command supports the following options:

`-a` *authenticationMethod*

Specify authentication method. The default value is what has been configured in the profile. The supported authentication methods are:

```
simple
sasl/CRAM-MD5
sasl/DIGEST-MD5
sasl/GSSAPI
tls:simple
tls:sasl/CRAM-MD5
tls:sasl/DIGEST-MD5
```

Selecting `simple` causes passwords to be sent over the network in clear text. Its use is strongly discouraged. Additionally, if the client is configured with a profile which uses no authentication, that is, either the `credentialLevel` attribute is set to `anonymous` or `authenticationMethod` is set to `none`, the user must use this option to provide an authentication method. If the authentication method is `sasl/GSSAPI`, *bindDN* and *bindPassword* is not required and the `hosts` and `ipnodes` fields of `/etc/nsswitch.conf` must be configured as:

```
hosts: dns files
ipnodes: dns files
```

See `nsswitch.conf(4)`.

`-b` *baseDN*
Create entries in the *baseDN* directory. *baseDN* is not relative to the client's default search base, but rather. it is the actual location where the entries will be created. If this parameter is not specified, the first search descriptor defined for the service or the default container will be used.

`-c`
Continue adding entries to the directory even after an error. Entries will not be added if the directory server is not responding or if there is an authentication problem.

`-D` *bindDN*
Create an entry which has write permission to the *baseDN*. When used with `-d` option, this entry only needs read permission.

`-d`
Dump the LDAP container to the standard output in the appropriate format for the given database.

`-f` *filename*
Indicates input file to read in an `/etc/` file format.

`-h` *LDAP_server*[:*serverPort*]
Specify an address (or a name) and an optional port of the LDAP server in which the entries will be stored. The current naming service specified in the `nsswitch.conf` file is used. The default value for the port is `389`, except when TLS is specified as the authentication method. In this case, the default LDAP server port number is `636`.

The format to specify the address and port number for an IPv6 address is:

`[`*ipv6_addr*`]:`*port*

To specify the address and port number for an IPv4 address, use the following format:

ipv4_addr`:`*port*

If the host name is specified, use the format:

host_name`:`*port*

`-j` *passwdFile*
Specify a file containing the password for the bind DN or the password for the SSL client's key database. To protect the password, use this option in scripts and place the password in a secure file. This option is mutually exclusive of the `-w` option.

`-M` *domainName*
The name of a domain served by the specified server. If not specified, the default domain name will be used.

-N *profileName*
Specify the DUAProfile name. A profile with such a name is supposed to exist on the server specified by -h option. Otherwise, a default DUAProfile will be used. The default value is default.

-P *certifPath*
The certificate path for the location of the certificate database. The value is the path where security database files reside. This is used for TLS support, which is specified in the authenticationMethod and serviceAuthenticationMethod attributes. The default is /var/ldap.

-p
Process the password field when loading password information from a file. By default, the password field is ignored because it is usually not valid, as the actual password appears in a shadow file.

-w *bindPassword*
Password to be used for authenticating the *bindDN*. If this parameter is missing, the command will prompt for a password. NULL passwords are not supported in LDAP.

When you use -w *bindPassword* to specify the password to be used for authentication, the password is visible to other users of the system by means of the ps command, in script files or in shell history.

If you supply "-" (hyphen) as a password, you will be prompted to enter a password.

-v
Verbose.

オペランド

The following operands are supported:

database
The name of the database or service name. Supported values are: aliases, auto_*, bootparams, ethers, group, hosts (including IPv6 addresses), netgroup, netmasks, networks, passwd, shadow, protocols, publickey, rpc, and services. Also supported are auth_attr, prof_attr, exec_attr, user_attr, and projects.

使用例

例 1 Adding Password Entries to the Directory Server

The following example shows how to add password entries to the directory server:

```
example# ldapaddent -D "cn=directory manager" -w secret \
      -f /etc/passwd passwd
```

例 2 Adding Group Entries

The following example shows how to add group entries to the directory server using sasl/CRAM-MD5 as the authentication method:

```
example# ldapaddent -D "cn=directory manager" -w secret \
     -a "sasl/CRAM-MD5" -f /etc/group group
```

例 3 Adding auto_master Entries

The following example shows how to add `auto_master` entries to the directory server:

```
example# ldapaddent -D "cn=directory manager" -w secret \
     -f /etc/auto_master auto_master
```

例 4 Dumping passwd Entries from the Directory to File

The following example shows how to dump `password` entries from the directory to a file `foo`:

```
example# ldapaddent -d passwd > foo
```

例 5 Adding Password Entries to a Specific Directory Server

The following example shows how to add password entries to a directory server that you specify:

```
example# ldapaddent -h 10.10.10.10:3890 \
-M another.domain.name -N special_duaprofile \
-D "cn=directory manager" -w secret \
-f /etc/passwd passwd
```

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

ファイル

`/var/ldap/ldap_client_file`
`/var/ldap/ldap_client_cred`
: Files containing the LDAP configuration of the client. These files are not to be modified manually. Their content is not guaranteed to be human readable. Use `ldapclient(1M)` to update these files.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis
Interface Stability	Committed

関連項目

`ldaplist(1)`, `ldapmodify(1)`, `ldapmodrdn(1)`, `ldapsearch(1)`, `idsconfig(1M)`, `ldapclient(1M)`, `nsswitch.conf(4)`, `attributes(5)`

『Oracle Solaris の管理: セキュリティーサービス』

注意

Currently StartTLS is not supported by `libldap.so.5`, therefore the port number provided refers to the port used during a TLS open, rather than the port used as part of a StartTLS sequence. For example:

```
-h foo:1000 -a tls:simple
```

The preceding refers to a raw TLS open on host `foo` port 1000, not an open, StartTLS sequence on an unsecured port 1000. If port 1000 is unsecured the connection will not be made.

名前

ldap_cachemgr – LDAP daemon to manage client configuration for LDAP based Network Information Service lookups

形式

`/usr/lib/ldap/ldap_cachemgr [-l log-file] [-g]`

機能説明

The `ldap_cachemgr` daemon is a process that provides an up-to-date configuration cache for LDAP naming services. It is started during multi-user boot.

The `ldap_cachemgr` utility provides caching for all parameters as specified and used by the LDAP naming service clients. The `ldap_cachemgr` utility uses the cache files which are originally created by executing the `ldapclient(1M)` utility, as cold start files. Updates to the cache files take place dynamically if profiles are used to configure the client. See the `init` option to `ldapclient(1M)`.

The `ldap_cachemgr` utility helps improve the performance of the clients that are using LDAP as the Naming service repository. In order for the LDAP naming services to function properly, the `ldap_cachemgr` daemon must be running. `ldap_cachemgr` also improves system security by making the configuration files readable by superuser only.

The cache maintained by this daemon is shared by all the processes that access LDAP Naming information. All processes access this cache through a door call. On startup, `ldap_cachemgr` initializes the cache from the cache files. See `ldapclient(1M)`. Thus, the cache survives machine reboots.

The `ldap_cachemgr` daemon also acts as its own administration tool. If an instance of `ldap_cachemgr` is already running, commands are passed transparently to the running version.

オプション

The following options are supported:

`-g`
: Print current configuration and statistics to standard output. This is the only option executable without superuser privileges.

`-l` *log-file*
: Cause `ldap_cachemgr` to use a log file other than the default `/var/ldap/cachemgr.log`.

使用例

例 1 Stopping and Restarting the ldap_cachemgr Daemon

The following example shows how to stop and to restart the `ldap_cachemgr` daemon.

```
example# svcadm disable network/ldap/client
example# svcadm enable network/ldap/client
```

例 2 Forcing ldap_cachemgr to Reread Configuration Files

The following example shows how to force `ldap_cachemgr` to reread the `/var/ldap/ldap_client_file` and `/var/ldap/ldap_client_cred` files

```
example# pkill -HUP ldap_cachemgr
```

ファイル

`/var/ldap/cachemgr.log`
: Default log file.

`/var/ldap/ldap_client_file`
`/var/ldap/ldap_client_cred`
: Files containing the LDAP configuration of the client. These files are not to be modified manually. Their content is not guaranteed to be human readable. Use `ldapclient(1M)` to update these files.

警告

The `ldap_cachemgr` utility is included in the current Solaris release on an uncommitted basis only. It is subject to change or removal in a future minor release.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`ldapadd(1)`, `ldapdelete(1)`, `ldaplist(1)`, `ldapmodify(1)`, `ldapmodrdn(1)`, `ldapsearch(1)`, `pkill(1)`, `svcs(1)`, `idsconfig(1M)`, `ldapaddent(1M)`, `ldapclient(1M)`, `svcadm(1M)`, `signal.h(3HEAD)`, `resolv.conf(4)`, `attributes(5)`, `ldap(5)`, `smf(5)`

注意事項

The `ldap_cachemgr` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/ldap/client
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

ldapclient – initialize LDAP client machine or output an LDAP client profile in LDIF format

形式

```
/usr/sbin/ldapclient [-v | -q] init [-a profileName=profileName]
     [-a domainName=domain] [-a proxyDN=proxyDN]
     [-a proxyPassword=password]
     [-a authenticationMethod=authenticationMethod]
     [-a enableShadowUpdate=true | false]
     [-a adminDN=adminDN]
     [-a adminPassword=adminPassword]
     [-a certificatePath=path] [-d bindDN] [-w bindPassword]
     [-j passwdFile] [-y passwdFile]
     [-z adminrPasswdFile] LDAP_server[:port_number]

/usr/sbin/ldapclient [-v | -q] manual [-a attrName=attrVal]

/usr/sbin/ldapclient [-v | -q] mod [-a attrName=attrVal]

/usr/sbin/ldapclient [-v | -q] list

/usr/sbin/ldapclient [-v | -q] uninit

/usr/sbin/ldapclient [-v | -q] genprofile -a profileName=profileName
     [-a attrName=attrVal]
```

機能説明

The `ldapclient` utility can be used to:

- initialize LDAP client machines
- restore the network service environment on LDAP clients
- list the contents of the LDAP client cache in human readable format.

The `init` form of the `ldapclient` utility is used to initialize an LDAP client machine, using a profile stored on an LDAP server specified by `LDAP_server`. The LDAP client will use the attributes in the specified profile to determine the configuration of the LDAP client. Using a configuration profile allows for easy installation of LDAP client and propagation of configuration changes to LDAP clients. The `ldap_cachemgr(1M)` utility will update the LDAP client configuration when its cache expires by reading the profile. For more information on the configuration profile refer to IETF document A Configuration Schema for LDAP Based Directory User Agents.

The `manual` form of the `ldapclient` utility is used to initialize an LDAP client machine manually. The LDAP client will use the attributes specified on the command line. Any unspecified attributes will be assigned their default values. At least one server must be specified in the `defaultServerList` or the `preferredServerList` attributes.The `domainName` attribute must be specified if the client's `domainName` is not set.

The `mod` form of the `ldapclient` utility is used to modify the configuration of an LDAP client machine that was setup manually. This option modifies only those LDAP client configuration attributes specified on the command line. The `mod` option should only be used on LDAP clients that were initialized using the `manual` option.

Regardless of which method is used for initialization, if a client is to be configured to use a proxy `credentialLevel`, proxy credentials must be provided using `-a proxyDN=`*proxyDN* and `-a proxyPassword=`*proxyPassword* options. However, if `-a proxyPassword=`*proxyPassword* is not specified, `ldapclient` will prompt for it. Note that `NULL` passwords are not allowed in LDAP. If a self `credentialLevel` is configured, `authenticationMethod` must be `sasl/GSSAPI`.

Similarily, if a client is to be configured to enable shadow information update and use a proxy credentialLevel, administrator credentials must be provided using `-a adminDN=`*adminDN* and `-a adminPassword=`*adminPassword*. However, the shadow information update does not need the administrator credentials if a self `credentialLevel` is configured.

The naming service-specific configuration properties are stored in the `svc:/network/ldap/client` SMF service. Modifying the SMF properties directly is not advised. Use this tool instead.

Other configuration might be modified during installation. It will be backed up to `/var/ldap/restore`. The files that are typically modified during initialization are:

- `/etc/nsswitch.conf`
- `/etc/defaultdomain` (if it exists)
- `/var/yp/binding/`‘`domainname`‘ (for a NIS [YP] client)

`ldapclient` does not set up a client to resolve hostnames using DNS. It simply copies `/etc/nsswitch.ldap` to `/etc/nsswitch.conf`. If you prefer to use DNS for host resolution, please refer to the DNS documentation for information on setting up DNS. See `resolv.conf(4)`. If you want to use `sasl/GSSAPI` as the authentication method, you have to use DNS for `hosts` and `ipnodes` resolution.

The `list` form of the `ldapclient` utility is used to list the LDAP client configuration. The output will be human readable. LDAP configuration files are not guaranteed to be human readable. Note that for security reason, the values for adminDN and adminPassword will not be displayed.

The `uninit` form of the `ldapclient` utility is used to uninitialize the network service environment, restoring it to the state it was in prior to the last execution of `ldapclient` using `init` or `manual`. The restoration will succeed only if the machine was initialized with the `init` or `manual` form of `ldapclient`, as it uses the backup files created by these options.

The `genprofile` option is used to write an LDIF formatted configuration profile based on the attributes specified on the command line to standard output. This profile can then be loaded into an LDAP server to be used as the client profile, which can be downloaded by means of the `ldapclient init` command. Loading the LDIF formatted profile to the directory server can be done through `ldapadd(1)`, or through any server specific import tool. Note that the attributes `proxyDN`, `proxyPassword`, `certificatePath`, `domainName`, `enableShadowUpdate`, `adminDN`, and `adminPassword` are not part of the configuration profile and thus are not permitted.

You must have superuser privileges to run the `ldapclient` command, except with the `genprofile` option.

To access the information stored in the directory, clients can either authenticate to the directory, or use an unauthenticated connection. The LDAP client is configured to have a credential level of either `anonymous` or `proxy`. In the first case, the client does not authenticate to the directory. In the second case, client authenticates to the directory using a proxy identity for read access, and using a administrator identity for write access if `enableShadowUpdate` is configured. In the third case, client authenticates to the directory using a Kerberos principal that is mapped to an LDAP identity by the LDAP server. Refer to the chapter on implementing security in the 『Oracle Solaris Administration: Naming and Directory Services』 or your appropriate directory server documentation for identity mapping details.

If a client is configured to use an identity, you can configure which authentication method the client will use. The LDAP client supports the following authentication methods:

```
none
simple
sasl/CRAM-MD5
sasl/DIGEST-MD5
sasl/GSSAPI
tls:simple
tls:sasl/CRAM-MD5
tls:sasl/DIGEST-MD5
```

Note that some directory servers may not support all of these authentication methods. For `simple`, be aware that the bind password will be sent in the clear to the LDAP server. For those authentication methods using TLS (transport layer security), the entire session is encrypted. You will need to install the appropriate certificate databases to use TLS.

Commands

The following commands are supported:

`init`
: Initialize client from a profile on a server.

`manual`
: Manually initialize client with the specified attribute values.

`mod`
: Modify attribute values in the configuration file after a manual initialization of the client.

`list`
: Write the contents of the LDAP client cache to standard output in human readable form.

`uninit`
: Uninitialize an LDAP client, assuming that `ldapclient` was used to initialize the client.

genprofile
: Generate a configuration profile in LDIF format that can then be stored in the directory for clients to use, with the `init` form of this command.

Attributes

The following attributes are supported:

adminDN
: Specify the Bind Distinguished Name for the administrator identity that is used for shadow information update. This option is required if the credential level is `proxy`, and `enableShadowUpdate` is set to `true`. There is no default value.

adminPassword
: Specify the administrator password. This option is required if the credential level is `proxy`, and `enableShadowUpdate` is set to `true`. There is no default value.

attributeMap
: Specify a mapping from an attribute defined by a service to an attribute in an alternative schema. This can be used to change the default schema used for a given service. The syntax of `attributeMap` is defined in the profile IETF draft. This option can be specified multiple times. The default value for all services is `NULL`. In the example,

```
attributeMap: passwd:uid=employeeNumber
```

the LDAP client would use the LDAP attribute `employeeNumber` rather than `uid` for the `passwd` service. This is a multivalued attribute.

authenticationMethod
: Specify the default authentication method used by all services unless overridden by the `serviceAuthenticationMethod` attribute. Multiple values can be specified by using a semicolon-separated list. The default value is `none`. For those services that use `credentialLevel` and `credentialLevel` is `anonymous`, this attribute is ignored. Services such as `pam_ldap` will use this attribute, even if `credentialLevel` is anonymous. The supported authentication methods are described above. If the authenticationMethod is `sasl/GSSAPI`, the `hosts` and `ipnodes` of `/etc/nsswitch.conf` must be configured with DNS support, for example:

```
hosts: dns files
ipnodes: dns files
```

bindTimeLimit
: The maximum time in seconds that a client should spend performing a bind operation. Set this to a positive integer. The default value is 30.

certificatePath
: The certificate path for the location of the certificate database. The value is the path where security database files reside. This is used for TLS support, which is specified in the `authenticationMethod` and `serviceAuthenticationMethod` attributes. The default is `/var/ldap`.

`credentialLevel`
Specify the credential level the client should use to contact the directory. The credential levels supported are either `anonymous` or `proxy`. If a `proxy` credential level is specified, then the `authenticationMethod` attribute must be specified to determine the authentication mechanism. Also, if the credential level is `proxy` and at least one of the authentication methods require a bind DN, the `proxyDN` and `proxyPassword` attribute values must be set. In addition, if `enableShadowUpdate` is set to `true`, the `adminDN` and `adminPassword` values must be set. If a self credential level is specified, the `authenticationMethod` must be `sasl/GSSAPI`.

`defaultSearchBase`
Specify the default search base DN. There is no default. The `serviceSearchDescriptor` attribute can be used to override the `defaultSearchBase` for given services.

`defaultSearchScope=one | sub`
Specify the default search scope for the client's search operations. This default can be overridden for a given service by specifying a `serviceSearchDescriptor`. The default is one level search.

`defaultServerList`
A space separated list of server names or server addresses, either IPv4 or IPv6. If you specify server names, be sure that the LDAP client can resolve the name without the LDAP name service. You must resolve the LDAP servers' names by using either `files` or `dns`. If the LDAP server name cannot be resolved, your naming service will fail.

The port number is optional. If not specified, the default LDAP server port number 389 is used, except when TLS is specified in the authentication method. In this case, the default LDAP server port number is 636.

The format to specify the port number for an IPv6 address is:

```
[ipv6_addr]:port
```

To specify the port number for an IPv4 address, use the following format:

```
ipv4_addr:port
```

If the host name is specified, use the format:

```
host_name:port
```

If you use TLS, the LDAP server's hostname must match the hostname in the TLS certificate. Typically, the hostname in the TLS certificate is a fully qualified domain name. With TLS, the LDAP server host addresses must resolve to the hostnames in the TLS certificate. You must use `files` or `dns` to resolve the host address.

`domainName`
Specify the DNS domain name. This becomes the default domain for the machine. The default is the current domain name. This attribute is only used in client initialization.

`enableShadowUpdate=true | false`
: Specify whether the client is allowed to update shadow information. If set to `true` and the credential level is `proxy`, `adminDN` and `adminPassword` must be specified.

`followReferrals=true | false`
: Specify the referral setting. A setting of true implies that referrals will be automatically followed and false would result in referrals not being followed. The default is true.

`objectclassMap`
: Specify a mapping from an `objectclass` defined by a service to an `objectclass` in an alternative schema. This can be used to change the default schema used for a given service. The syntax of `objectclassMap` is defined in the profile IETF draft. This option can be specified multiple times. The default value for all services is `NULL`. In the example,

```
objectclassMap=passwd:posixAccount=unixAccount
```

the LDAP client would use the LDAP `objectclass` of `unixAccount` rather than the `posixAccount` for the `passwd` service. This is a multivalued attribute.

`preferredServerList`
: Specify the space separated list of server names or server addresses, either IPv4 or IPv6, to be contacted before servers specified by the `defaultServerList` attribute. If you specify server names, be sure that the LDAP client can resolve the name without the LDAP name service. You must resolve the LDAP servers' names by using either `files` or `dns`. If the LDAP server name cannot be resolved, your naming service will fail.

The port number is optional. If not specified, the default LDAP server port number 389 is used, except when TLS is specified in the authentication method. In this case, the default LDAP server port number is 636.

The format to specify the port number for an IPv6 address is:

```
[ipv6_addr]:port
```

To specify the port number for an IPv4 address, use the following format:

```
ipv4_addr:port
```

If the host name is specified, use the format:

```
host_name:port
```

If you use TLS, the LDAP server's hostname must match the hostname in the TLS certificate. Typically, the hostname in the TLS certificate is a fully qualified domain name. With TLS, the LDAP server host addresses must resolve to the hostnames in the TLS certificate. You must use `files` or `dns` to resolve the host address.

`profileName`
: Specify the profile name. For `ldapclient init`, this attribute is the name of an existing profile which may be downloaded periodically depending on the value of the `profileTTL` attribute. For `ldapclient genprofile`, this is the name of the profile to be generated. The default value is `default`.

`profileTTL`
Specify the TTL value in seconds for the client information. This is only relevant if the machine was initialized with a client profile. If you do not want `ldap_cachemgr(1M)` to attempt to refresh the LDAP client configuration from the LDAP server, set `profileTTL` to 0 (zero). Valid values are either zero 0 (for no expiration) or a positive integer in seconds. The default value is 12 hours.

`proxyDN`
Specify the Bind Distinguished Name for the proxy identity. This option is required if the credential level is `proxy`, and at least one of the authentication methods requires a bind DN. There is no default value.

`proxyPassword`
Specify client proxy password. This option is required if the credential level is `proxy`, and at least one of the authentication methods requires a bind DN. There is no default.

`searchTimeLimit`
Specify maximum number of seconds allowed for an LDAP search operation. The default is 30 seconds. The server may have its own search time limit.

`serviceAuthenticationMethod`
Specify authentication methods to be used by a service in the form *servicename*:`authenticationmethod`, for example:

```
pam_ldap:tls:simple
```

For multiple authentication methods, use a semicolon-separated list. The default value is no service authentication methods, in which case, each service would default to the `authenticationMethod` value. The supported authentications are described above.

Three services support this feature: `passwd-cmd`, `keyserv`, and `pam_ldap`. The `passwd-cmd` service is used to define the authentication method to be used by `passwd(1)` to change the user's password and other attributes. The `keyserv` service is used to identify the authentication method to be used by the `chkey(1)` and `newkey(1M)` utilities. The `pam_ldap` service defines the authentication method to be used for authenticating users when `pam_ldap(5)` is configured. If this attribute is not set for any of these services, the `authenticationMethod` attribute is used to define the authentication method. This is a multivalued attribute.

`serviceCredentialLevel`
Specify credential level to be used by a service. Multiple values can be specified in a space-separated list. The default value for all services is `NULL`. The supported credential levels are: `anonymous` or `proxy`. At present, no service uses this attribute. This is a multivalued attribute.

`serviceSearchDescriptor`
Override the default base DN for LDAP searches for a given service. The format of the descriptors also allow overriding the default search scope and search filter for each service.

The syntax of `serviceSearchDescriptor` is defined in the profile IETF draft. The default value for all services is `NULL`. This is a multivalued attribute. In the example,

```
serviceSearchDescriptor=passwd:ou=people,dc=a1,dc=acme,dc=com?one
```

the LDAP client would do a one level search in `ou=people,dc=a1,dc=acme,dc=com` rather than `ou=people,`*defaultSearchBase* for the `passwd` service.

オプション

The following options are supported:

`-a attrName=`*attrValue*
Specify `attrName` and its value. See `SYNOPSIS` for a complete list of possible attribute names and values.

`-D` *bindDN*
Specifies an entry that has read permission for the requested database.

`-j` *passwdFile*
Specify a file containing the password for the bind DN or the password for the SSL client's key database. To protect the password, use this option in scripts and place the password in a secure file. This option is mutually exclusive of the `-w` option.

`-q`
Quiet mode. No output is generated.

`-v`
Verbose output.

`-w` *bindPassword*
Password to be used for authenticating the bind DN. If this parameter is missing, the command will prompt for a password. `NULL` passwords are not supported in LDAP.

When you use `-w` *bindPassword* to specify the password to be used for authentication, the password is visible to other users of the system by means of the `ps` command, in script files, or in shell history.

If you supply "`-`" (hyphen) as a password, the command will prompt for a password.

`-y` *passwdFile*
Specify a file containing the password for the proxy DN. To protect the password, use this option in scripts and place the password in a secure file. This option is mutually exclusive of the `-a` *proxyPassword* option.

`-z` *adminrPasswdFile*
Specify a file containing the password for the `adminDN`. To protect the password, use this option in scripts and place the password in a secure file. This option is mutually exclusive of the `-a` *adminPassword* option.

オペランド

The following operand is supported:

LDAP_server

An address or a name for the LDAP server from which the profile will be loaded. The current naming service specified in the `nsswitch.conf` file is used. Once the profile is loaded, the `preferredServerList` and `defaultServerList` specified in the profile are used.

使用例

例 1 Setting Up a Client By Using the Default Profile Stored on a Specified LDAP Server

The following example shows how to set up a client using the default profile stored on the specified LDAP server. This command will only be successful if either the credential level in the profile is set to `anonymous` or the authentication method is set to `none`.

```
example# ldapclient init 172.16.100.1
```

例 2 Setting Up a Client By Using the simple Profile Stored on a Specified LDAP Server

The following example shows how to set up a client using the `simple` profile stored on the specified LDAP server. The domainname is set to `xyz.mycompany.com` and the proxyPassword is `secret`.

```
example# ldapclient init -a profileName=simple \
-a domainName=xyz.mycompany.com \
-a proxyDN=cn=proxyagent,ou=profile,dc=xyz,dc=mycompany,dc=com \
-a proxyPassword=secret '['fe80::a00:20ff:fea3:388']':386
```

例 3 Setting Up a Client Using Only One Server

The following example shows how to set up a client using only one server. The authentication method is set to `none`, and the search base is `dc=mycompany,dc=com`.

```
example# ldapclient manual -a authenticationMethod=none \
-a defaultSearchBase=dc=mycompany,dc=com \
-a defaultServerList=172.16.100.1
```

例 4 Setting Up a Client Using Only One Server That Does Not Follow Referrals

The following example shows how to set up a client using only one server. The credential level is set to `proxy`. The authentication method of is `sasl/CRAM-MD5`, with the option not to follow referrals. The domain name is `xyz.mycompany.com`, and the LDAP server is running on port number 386 at IP address `172.16.100.1`.

```
example# ldapclient manual \
-a credentialLevel=proxy \
-a authenticationMethod=sasl/CRAM-MD5 \
-a proxyPassword=secret \
-a proxyDN=cn=proxyagent,ou=profile,dc=xyz,dc=mycompany,dc=com \
-a defaultSearchBase=dc=xyz,dc=mycompany,dc=com \
-a domainName=xyz.mycompany.com \
-a followReferrals=false \
```

例 4 Setting Up a Client Using Only One Server That Does Not Follow Referrals *(続き)*

```
-a defaultServerList=172.16.100.1:386
```

例 5 Using genprofile to Set Only the defaultSearchBase and the Server Addresses

The following example shows how to use the `genprofile` command to set the `defaultSearchBase` and the server addresses.

```
example# ldapclient genprofile -a profileName=myprofile \
-a defaultSearchBase=dc=eng,dc=sun,dc=com \
-a "defaultServerList=172.16.100.1 172.16.234.15:386" \
> myprofile.ldif
```

例 6 Creating a Profile on IPv6 servers

The following example creates a profile on IPv6 servers

```
example# ldapclient genprofile -a profileName=eng \
-a credentialLevel=proxy \
-a authenticationMethod=sasl/DIGEST-MD5 \
-a defaultSearchBase=dc=eng,dc=acme,dc=com \
-a "serviceSearchDescriptor=passwd:ou=people,dc=a1,dc=acme,dc=com?one"\
-a preferredServerList= '['fe80::a00:20ff:fea3:388']' \
-a "defaultServerList='['fec0::111:a00:20ff:fea3:edcf']' \
    '['fec0::111:a00:20ff:feb5:e41']'" > eng.ldif
```

例 7 Creating a Profile That Overrides Every Default Value

The following example shows a profile that overrides every default value.

```
example# ldapclient genprofile -a profileName=eng \
-a credentialLevel=proxy -a authenticationMethod=sasl/DIGEST-MD5 \
-a bindTimeLimit=20 \
-a defaultSearchBase=dc=eng,dc=acme,dc=com \
-a "serviceSearchDescriptor=passwd:ou=people,dc=a1,dc=acme,dc=com?one"\
-a serviceAuthenticationMethod=pam_ldap:tls:simple \
-a defaultSearchScope=sub \
-a attributeMap=passwd:uid=employeeNumber \
-a objectclassMap=passwd:posixAccount=unixAccount \
-a followReferrals=false -a profileTTL=6000 \
-a preferredServerList=172.16.100.30 -a searchTimeLimit=30 \
-a "defaultServerList=172.16.200.1 172.16.100.1 192.168.5.6" > eng.ldif
```

終了ステータス

The following exit values are returned:

0 The command successfully executed.

1 An error occurred. An error message is output.

2 `proxyDN` and `proxyPassword` attributes are required, but they are not provided.

ファイル

`/var/ldap/ldap_client_cred`
`/var/ldap/ldap_client_file`
Contain the LDAP configuration of the client. These files are not to be modified manually. Their content is not guaranteed to be human readable. Use `ldapclient` to update them.

`/etc/defaultdomain`
System default domain name, matching the domain name of the data in the LDAP servers. See `defaultdomain(4)`.

`/etc/nsswitch.conf`
Configuration file for the name-service switch. See `nsswitch.conf(4)`.

`/etc/nsswitch.ldap`
Sample configuration file for the name-service switch configured with LDAP and files.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis
Interface Stability	Committed

関連項目

`chkey(1)`, `ldapadd(1)`, `ldapdelete(1)`, `ldaplist(1)`, `ldapmodify(1)`, `ldapmodrdn(1)`, `ldapsearch(1)`, `idsconfig(1M)`, `ldapaddent(1M)`, `ldap_cachemgr(1M)`, `defaultdomain(4)`, `nsswitch.conf(4)`, `resolv.conf(4)`, `attributes(5)`

注意

Currently `StartTLS` is not supported by `libldap.so.5`, therefore the port number provided refers to the port used during a TLS open, rather than the port used as part of a `StartTLS` sequence. To avoid timeout delays, mixed use of TLS and non-TLS authentication mechanisms is not recommended.

For example:

```
-h foo:1000 -a authenticationMethod=tls:simple
```

...or:

```
defaultServerList= foo:1000
authenticationMethod= tls:simple
```

The preceding refers to a raw TLS open on host `foo` port 1000, not an open, StartTLS sequence on an unsecured port 1000. If port 1000 is unsecured the connection will not be made.

As a second example, the following will incur a significant timeout delay while attempting the connection to `foo:636` with an unsecured bind.

```
defaultServerList= foo:636 foo:389
authenticationMethod= simple
```

名前

ldmad – Logical Domains Agents daemon

形式

/usr/lib/ldoms/ldmad

機能説明

The ldmad daemon is part of the framework that enables Logical Domain agents to run on a Logical Domain. A Logical Domain agent is a component that interacts with the control domain to provide features or information.

ldmad is responsible for running agents on a Logical Domain. ldmad must be enabled to ensure the proper functionality of all features provided by the domain manager on the control domain. The daemon is started at boot time and has no configuration options.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/ldoms
Interface Stability	Unstable

関連項目

svcs(1), svcadm(1M), syslog(3C), syslog.conf(4), attributes(5), smf(5)

『Oracle VM Server for SPARC 2.1 管理ガイド』

エラー

ldmad uses syslog(3C) to report status and error messages. Error messages are logged with the LOG_ERR and LOG_NOTICE priorities. Informational messages are logged with the LOG_INFO priority. The default entries in the /etc/syslog.conf file specify that all ldmad error messages are written to the /var/adm/messages log.

注意事項

The ldmad service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/ldoms/agents:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

link, unlink – link and unlink files and directories

形式

`/usr/sbin/link` *existing-file* *new-file*

`/usr/xpg4/bin/link` *existing-file* *new-file*

`/usr/sbin/unlink` *file*

機能説明

The `link` and `unlink` commands link and unlink files and directories. Only super-users can use these commands on directories.

Use `link` to create a new file that points to an existing file. The *existing-file* and *new-file* operands specify the existing file and newly-created files. See `OPERANDS`.

Note that the ZFS file system does not support links between directories.

`link` and `unlink` directly invoke the `link(2)` and `unlink(2)` system calls, performing exactly what they are told to do and abandoning all error checking. This differs from the `ln(1)` command. See `ln(1)`.

While linked files and directories can be removed using `unlink`, it is safer to use `rm(1)` and `rmdir(1)` instead. See `rm(1)` and `rmdir(1)`.

/usr/xpg4/bin/link

If the existing file being hard linked is itself a symbolic link, then the newly created file (*new-file*) will be a hard link to the file referenced by the symbolic link, not to the symbolic link object itself (*existing-file*).

オペランド

The following operands are supported:

existing-file Specifies the name of the existing file to be linked.

file Specifies the name of the file to be unlinked.

new-file Specifies the name of newly created (linked) file.

環境変数

See `environ(5)` for descriptions of the following environment variables that affect the execution of `link`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

/usr/xpg4/bin/link

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See `standards(5)`.

関連項目 ln(1), rm(1), link(2), unlink(2), attributes(5), environ(5), standards(5)

名前

llc2_loop – loopback diagnostics to test the driver, adapter and network.

形式

```
/usr/lib/llc2/llc2_loop2 [-v] ppa
/usr/lib/llc2/llc2_loop3 ppa sap frames
/usr/lib/llc2/llc2_loop3 ppa type frames
/usr/lib/llc2/llc2_loop4 [-v] ppa
```

機能説明

Loop 2

The `loop2` test sends a NULL XID frame to the broadcast (all 1's) destination MAC address. The source SAP (Service Access Point) value used is 0x04 (SNA's SAP). Therefore, if SNA is running on the system, the `loop2` test will fail. The destination SAP value is the NULL SAP (0x00). This test finds out who is listening and can receive frames sent out from a node. The verbose (`-v`) option displays the MAC address of responding nodes. All possible responders may not be displayed, since the `loop2` test only waits for responses for 2 seconds, but during this time 50-200 nodes may be displayed. The most likely error is:

```
Unexpected DLPI primitive x, expected y.
```

where $x = 5$ and $y = 6$. From `/usr/include/sys/dlpi.h`, the expected return value from one of the DLPI primitives is 6 (`DL_OK_ACK`), but instead a 5 (`DL_ERROR_ACK`) was received. This can occur for two reasons:

- The `loop2` command was issued to a non-existent PPA (Physical Point of Attachment).
- The SAP (0x04) is already in use (for example, the SNA subsystem is up).

Loop 3

The `loop3` test sends 1,495 byte Unnumbered Information (UI) frames to the NULL (all 0's) destination MAC address. This should be used along with data capture either on the local node or another node on the same LAN to verify the transmission of data. The *ppa* argument specifies the adapter on which to run the test. The *ppa* is the relative physical position of the adapter and may be ascertained by viewing the adapter configuration (see `llc2_config(1)`). For Token Ring or Ethernet, specify an even *sap* value from 2 through 254, or, for Ethernet only, any *type* value from 1519 (0x05ef) through 65535 (0xffff). It is advised to pick a value that is easily recognized when the data capture output is viewed. *frames* is the decimal number of 1,495 bytes packets to transmit. The test will only display a message if a failure occurs.

Loop 4

The `loop4` test sends a TEST frame (no information field) to the broadcast (all 1's) destination MAC address. The source SAP value used is 0x04 (SNA's SAP). Therefore, if SNA is running on the system, the `loop4` test will fail. The destination SAP value is the NULL SAP (0x00). This test finds out who is listening and can receive frames sent out from a node. The verbose (`-v`) option displays the MAC address of responding nodes. All possible responders may not be displayed since the `loop4` test only waits for responses for 2 seconds, but during this time 50-200 nodes may be displayed. The `loop4` test displays information similar to the following example if other nodes are listening and respond (verbose mode):

```
-Attaching
    -Binding
```

```
    -Sending TEST
    -Responders
       1-0000c0c12449
       2-08000e142990
       3-08000e142a51
       4-0000c0450044
       5-0000c0199e46
    -Unbinding
-Detaching
5 nodes responding
```

The errors displayed are the same as for `loop2`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/llc2

関連項目

`llc2_config(1)`, `llc2(4)`, `attributes(5)`, `llc2(7D)`

注意事項

For information about how to start the service, see `llc2(7D)`

名前

lldpadm – Link Layer Discovery Protocol administration tool

形式

```
lldpadm set-agentprop -p prop[+|-]=value[,...] lldp_agent
lldpadm reset-agentprop -p prop[,...] lldp_agent
lldpadm show-agentprop [[-c] -o field,...] -p prop[,...] [lldp_agent]

lldpadm set-tlvprop -p prop[+|-]=value[,...] tlv_name
lldpadm reset-tlvprop -p prop[,...] tlv_name
lldpadm show-tlvprop [[-c] -o field,...] -p prop[,...] [tlv_name]

lldpadm set-agenttlvprop -p prop[+|-]=value[,...] -a lldp_agent
     tlv_name
lldpadm reset-agenttlvprop -p prop[,...] -a lldp_agent tlv_name
lldpadm show-agenttlvprop [[-c] -o field,...] -p prop[,...]
     [-a lldp_agent] [tlv_name]

lldpadm show-agent [-c] [-s] [-v] -o field,...] [-l|-r] [lldp_agent]
```

機能説明

The `lldpadm` command is used to enable or disable a Link Layer Discovery Protocol (LLDP) agent on a physical datalink. `lldpadm` is also used to configure the behavior of an LLDP agent. The LLDP agent implements the LLDP protocol for a given physical datalink. LLDP is a one-way link layer protocol that allows an IEEE 802 LAN station to advertise the capabilities and current status of the system to other stations attached to the same LAN. The LLDP agent can also receive information about the capabilities and current status of the system associated with a remote station. LLDP agent can either be enabled for transmission only, for reception only, or for both.

Information to be exchanged is packed as a sequence of type, length, and value (TLVs), wherein the type field identifies the type of information, the length field indicates the length of the information field in octets, and the value field contains the information itself.

オペランド

Each `lldpadm` subcommand operates on one of the following objects:

lldp_agent

An LLDP agent implements the LLDP protocol for a given physical datalink that is connected to IEEE 802 LAN. The only supported physical links are the ones of media type Ethernet. Thus, LLDP can be enabled on all the links displayed in `dladm show-phys` output that are of media type Ethernet. The name of the *lldp_agent* is the name of the datalink itself.

tlv_name

Name of the TLV whose value can be modified. The supported modifiable TLVs are:

- `syscapab` and `mgmtaddr`. These form Global TLVs that are common to all the LLDP agents on the system.
- `pfc` and `appln`. Per-LLDP agent TLVs.

サブコマンド

`lldpadm` supports the following subcommands.

`lldpadm set-agentprop|set-ap -p` *prop*[+|-]=*value*[,...] *lldp_agent*

Sets the value of one or more LLDP agent properties to the value specified. If the property takes multiple values then the value should be specified with a comma as the delimiter. The value is always made persistent and thus will be reapplied on system reboot or `lldpd(1M)` daemon restart. The list of properties supported and each property's possible values can be retrieved using `show-agentprop` subcommand.

`-p` *prop*[+|-]=*value*[,...], `--prop` *prop*[+|-]=*value*[,...]

A comma-separated list of properties to be set to the specified values. It also provides the following qualifiers to perform add and delete operations in addition to assignment.

`+` Adds the given value to the current list of value(s).

`-` Removes the given value from the current list of value(s).

`=` Makes a new assignment and removes all the current value(s).

See `EXAMPLES` for more information on how to use the qualifiers.

`lldpadm reset-agentprop|reset-ap -p` *prop*[,...] *lldp_agent*

Resets one or more properties to their default values. The default values for properties can be retrieved using `show-agentprop` subcommand.

`-p` *prop*[,...], `--prop` *prop*[,...]

A comma-separated list of properties to reset.

`lldpadm show-agentprop|show-ap` [[`-c`] `-o` *field*,...] `-p` *prop*[,...] [*lldp_agent*]

Show the current value of one or more properties, either for all of the LLDP agents or for the specified LLDP agent. Several properties of interest can be retrieved at a time by providing comma-separated property names to `-p` option. If the `-p` option is not specified, all available properties are displayed.

`-o` *field*[,...], `--output` *field*[,...]

A case-insensitive, comma-separated list of output fields to display. The field name must be one of the fields listed below, or the special value `all` to display all fields. For each LLDP agent, the following fields can be displayed:

`AGENT`

The name of the LLDP agent.

`PROPERTY`

The name of the property.

`PERM`

The read/write permissions of the property. The value shown will be `r` (read only), `w` (write only) or `rw` (read/write).

`VALUE`
The current value of the property. If the value is not set, it is shown as `--`. If it is unknown, the value is shown as `?`.

`DEFAULT`
The default value of the property. If the property has no default value, `--` is shown.

`POSSIBLE`
A comma-separated list of the values the property can have. If the values span a numeric range, *min* - *max* might be shown as shorthand. If the possible values are unknown or unbounded, `--` is shown.

`-c`, `--parsable`
Display using a stable machine-parsable format. The `-o` option is required with this option. See "Parsable Output Format", below.

`-p` *prop*[,...], `--prop` *prop*[,...]
A comma-separated list of properties to display.

For the supported list of agent properties, see "Agent Properties" section below.

`lldpadm set-tlvprop|set-tp -p` *prop*[+|-]=*value*[,...] *tlv_name*
Sets the value of one or more TLV properties to the value specified. If the property takes multiple values, the value should be specified with a comma as the delimiter. The value is always persisted and will be reapplied on system reboot or `lldpd(1M)` daemon restart. The list of properties supported and each property's possible values can be retrieved using `show-tlvprop` subcommand.

注 – The TLVs modified using this subcommand apply to all the LLDP agents running on the system.

`-p` *prop*[+|-]=*value*[,...], `--prop` *prop*[+|-]=*value*[,...]
See the description of this option under the `set-agentprop` subcommand, above.

`lldpadm reset-tlvprop|reset-tp -p` *prop*[,...] *tlv_name*
Resets one or more properties to their default values. The default values for properties can be retrieved using `show-tlvprop` subcommand.

`-p` *prop*[,...], `--prop` *prop*[,...]
A comma-separated list of properties to reset.

`lldpadm show-tlvprop|show-tp` [[`-c`] `-o` *field*,...] `-p` *prop*[,...] [*lldp_agent*]
Show the current value of one or more properties, either for all the TLVs or for a specified TLV. Several properties of interest can be retrieved at a time by providing comma-separated property names to `-p` option. If the `-p` option is not specified, all available properties are displayed.

`-o` *field*[,...], `--output` *field*[,...]
A case-insensitive, comma-separated list of output fields to display. The field name must be one of the fields listed below, or the special value `all` to display all fields. For each TLV, the following fields can be displayed:

`TLVNAME`
The name of the TLV.

`PROPERTY`
The name of the property.

`PERM`
The read/write permissions of the property. The value shown will be `r` (read only), `w` (write only) or `rw` (read/write).

`VALUE`
The current value of the property. If the value is not set, it is shown as `--`. If it is unknown, the value is shown as `?`.

`DEFAULT`
The default value of the property. If the property has no default value, `--` is shown.

`POSSIBLE`
A comma-separated list of the values the property can have. If the values span a numeric range, *min* - *max* might be shown as shorthand. If the possible values are unknown or unbounded, `--` is shown.

`-c`, `--parsable`
Display using a stable machine-parsable format. The `-o` option is required with this option. See "Parsable Output Format", below.

`-p` *prop*[,...], `--prop` *prop*[,...]
A comma-separated list of properties to display.

`lldpadm set-agenttlvprop`|`set-atp -p` *prop*[+|-]=*value*[,...] `-a` *lldp_agent tlv_name*
Sets the value of one or more TLV properties to the value specified. The `-a` option is mandatory and identifies the name of the agent for which the TLV property needs to be set. In this way, the TLV property modification is reflected only on the specified agent. If the property takes multiple values then the value should be specified with a comma as the delimiter. The value is always made persistent and will be reapplied on system reboot or `lldpd(1M)` daemon restart. The list of properties supported and each property's possible values can be retrieved using `show-agenttlvprop` subcommand.

`-p` *prop*[+|-]=*value*[,...], `--prop` *prop*[+|-]=*value*[,...]
See the description of this option under the `set-agentprop` subcommand, above.

`-a` *lldp_agent*, `--agent=`*lldp_agent*
The name of the LLDP agent for which TLV properties need to be displayed.

`lldpadm reset-agenttlvprop|reset-atp -p` *prop*[,...]=*value*[,...] `-a` *lldp_agent tlv_name*

Resets one or more properties to their default values. The `-a` option is mandatory and identifies the name of the agent for which the TLV property needs to be reset. The default values for properties can be retrieved using `show-agenttlvprop` subcommand.

`-p` *prop*[,...], `--prop` *prop*[,...]

A comma-separated list of properties to display.

`-a` *lldp_agent*, `--agent=`*lldp_agent*

The name of the LLDP agent for which TLV properties need to be displayed.

`lldpadm show-agenttlvprop|show-atp` [[`-c`] `-o` *field*,...] `-p` *prop*[,...] [`-a` *lldp_agent*] [*tlv_name*]

Show the current value of one or more properties, either for all of the TLVs or for a specified TLV. If a single LLDP agent is not specified (using `-a`), TLV properties for all LLDP agents are displayed. Several properties of interest can be retrieved at a time by providing comma-separated property names to the `-p` option. If the `-p` option is not specified, all available properties are displayed.

`-o` *field*[,...], `--output` *field*[,...]

A case-insensitive, comma-separated list of output fields to display. The field name must be one of the fields listed below, or the special value `all` to display all fields. For each LLDP agent, the following fields can be displayed:

`AGENT`

The name of the LLDP agent.

`TLVNAME`

The name of the TLV.

`PROPERTY`

The name of the property.

`PERM`

The read/write permissions of the property. The value shown will be `r` (read only), `w` (write only) or `rw` (read/write).

`VALUE`

The current value of the property. If the value is not set, it is shown as `--`. If it is unknown, the value is shown as `?`.

`DEFAULT`

The default value of the property. If the property has no default value, `--` is shown.

`POSSIBLE`

A comma-separated list of the values the property can have. If the values span a numeric range, *min* - *max* might be shown as shorthand. If the possible values are unknown or unbounded, `--` is shown.

`-c`, `--parsable`
Display using a stable machine-parsable format. The `-o` option is required with this option. See "Parsable Output Format", below.

`-p` *prop*[,...], `--prop` *prop*[,...]
A comma-separated list of properties to display.

`-a` *lldp_agent*, `--agent=`*lldp_agent*
The name of the LLDP agent for which TLV properties need to be displayed.

For the supported list of TLV properties that apply on a per-LLDP agent basis, see the "Per-LLDP agent TLV properties" section, below.

`lldpadm show-agent` [`-c`] [`-s`] [`-v`] `-o` *field*,...] [`-l` | `-r`] [*lldp_agent*]
Show the information advertised by the specified LLDP agent or information advertised by the adjacent neighbors to the specified LLDP agent. If no LLDP agent is specified, then the local or remote information will be displayed for all the LLDP agents. The information is displayed as a multi-line output, with each line containing information about a single TLV in the following format:

<Name of the TLV> : *<TLV Information expressed as a string>*

The following lines can be displayed:

```
                    Agent:
        Chassis ID Subtype:
                Chassis ID:
           Port ID Subtype:
                   Port ID:
          Port Description:
              Time to Live:
               System Name:
        System Description:
    Supported Capabilities:
      Enabled Capabilities:
        Management Address:
        Maximum Frame Size:
              Port VLAN ID:
              VLAN Name/ID:
       VNIC PortID/VLAN ID:
   Aggregation Information:
               PFC Willing:
                   PFC Cap:
                   PFC MBC:
                PFC Enable:
Application(s)(ID/Sel/Pri):
  Next Packet Transmission:  [displayed only when -l is used]
   Information Valid Until:  [displayed only when -r is used]
```

`-c`, `--parsable`
Display using a stable machine-parsable format. The `-o` option is required with this option. See "Parsable Output Format", below.

`-l`, `--local`
Displays information advertised by the local LLDP agent. This option is mutually exclusive of the `-r` option.

`-r`, `--remote`
Displays information advertised by the adjacent neighbors. This option is mutually exclusive of the `-l` option.

`-s`
Show the statistics for the specified LLDP agents or for all the LLDP agents on the system. Every LLDP agent maintains statistical counters that are used to count significant events in the transmit and receive state machines. These counters are defined to be 32-bit unsigned integers.

`-o` *field*[,...], `--output` *field*[,...]
A case-insensitive, comma-separated list of output fields to display. The field name must be one of the fields listed below, or the special value `all` to display all fields. For each TLV, the following fields can be displayed:

`AGENT`
The name of the LLDP agent.

`IFRAMES`
Count of all LLDP frames received by `AGENT`.

`IERR`
Count of all LLDPDUs received with one or more detectable errors.

`IDISCARD`
Count of all LLDPDUs received and then discarded for any of the following reasons:

- incorrectly formed LLDPDUs with respect to the first three mandatory TLVs
- insufficient space to store the incoming LLDPDU

`OFRAMES`
Count of all LLDP frames transmitted by `AGENT`.

`OLENERR`
Count of times the outgoing LLDPDU exceeded the length restrictions of 1500 bytes.

`TLVDISCARD`
Count of all TLVs received and then discarded because the TLVs did not adhere to the TLV usage rules as specified by the standard.

`TLVUNRECOG`
Count of all TLVs received that are not recognized by the LLDP agent.

`AGEOUT`
Count of the times that a neighbor's information has been deleted because of aging.

`-v`, `--verbose`
Displays detailed information.

Parsable Output Format

The `lldpadm show` subcommands have a `-c` option that displays output in a machine-parsable format. The output format is one or more lines of colon (:)-delimited fields. The fields displayed are specific to the subcommand used and are listed under the entry for the `-o` option for a given subcommand. Output includes only those fields requested by means of the `--o` option, in the order requested. Note that the `-o all` option, which displays all the fields for a given subcommand, cannot be used with the parsable output option.

When you request multiple fields, any literal colon characters are escaped by a backslash (`\`) before being output. Similarly, literal backslash characters are also escaped with a backslash. This escape format is parsable by using shell `read(1)` functions with the environment variable set as `IFS=`: Note that escaping is not done when you request only a single field.

Agent Properties

The following LLDP agent properties are supported:

`mode`
Configures the operation mode of the LLDP agent. Possible values are:

`txonly`
Enables LLDP for transmission only.

`rxonly`
Enables LLDP for receiving only.

`both`
Enables LLDP for both transmission and receiving.

`disable`
Disables LLDP on the LLDP agent.

Every LLDP packet (LLDPDU) transmitted by an LLDP agent contains multiple TLVs. The following four TLVs are mandatory and therefore included in all the LLDPDUs transmitted by an agent configured in the `txonly` or both mode:

`CHASSIS ID`
The value transmitted in the Chassis ID TLV is:

- subtype = 7 (locally assigned)
- Chassis ID = `hostid(1)`

`PORT ID`
The value transmitted in the Port ID TLV is:

- subtype = 3 (MAC address)
- Port ID = primary MAC address of the agent

`TTL`
The duration for which this packet is valid. The default value is 30 seconds.

`End of PDU TLV`
End of PDU indicator.

Optional TLVs that can be advertised are configured using the following properties:

`basic-tlv`
Configures the Basic Management TLVs that should be advertised by the LLDP agent. The possible values are:

`portdesc`
Alphanumeric string that identifies the datalink. Value set to the linkname.

`sysname`
Alphanumeric string that identifies the system. Value set to the output of `'uname -n'`.

`sysdesc`
Alphanumeric string that describes the system. Value set to the output of `'uname -a'`.

`syscapab`
Indicates the systems supported and enabled capabilities.

`mgmtaddr`
Indicates the IP address of the system that can be used by network management.

`dot1-tlv`
Configures the IEEE 802.1 Organizationally Specific TLVs that should be advertised by the LLDP agent. The possible values are:

`vlanname`
Indicates the names and IDs of all the VLANS configured on the datalink.

`pvid`
Indicates the default VLAN ID associated with the given datalink. It corresponds to the `default_tag` datalink property that is managed by means of the `dladm(1M)` utility.

`linkaggr`
Indicates whether underlying datalink is in an aggregation or is capable of being part of an aggregation.

`pfc`
Indicates whether underlying datalink supports PFC (Priority Flow Control) and the priorities for which the PFC pause frame is enabled. Also indicates whether the local endpoint is willing to negotiate the PFC configuration.

`appln`
Indicates the priority that will be used by an application.

`dot3-tlv`
Configures the IEEE 802.3 Organizationally Specific TLVs that should be advertised by the LLDP agent. The possible values are:

`max-framesize`
Indicates the maximum supported frame size for the underlying datalink.

`virt-tlv`
Configures the Solaris's Virtualization Specific TLVs that should be advertised by the LLDP agent. The possible values are:

`vnic`
Indicates the MAC address of the Virtual NIC created on top of the underlying physical link. Also indicates any VLAN id associated with the VNIC. See `dladm(1M)` for more information on VNIC.

Global TLVs and Their Properties

The following Global TLV properties are supported:

`syscapab` (TLV name)
This property can one of the following values:

`supported`
Indicates the supported capabilities on the system. The default supported capabilities are: `bridge`, `router`, and `station`.

`enabled`
Indicates the enabled capabilities on the system. The enabled capabilities must be a subset of the supported capabilities.

`mgmtaddr` (TLV name)
This property can have the following value:

`ipaddr`
The IP address(es), either IPv4 or IPv6, associated with the local LLDP agent that will be used to reach higher layer entities to assist discovery by network management.

Per-Agent TLVs and Their Properties

The following Agent TLV properties are supported:

`appln` (TLV name)
This property can have the following value:

`apt`
Configures the Application Priority Table for an Application TLV. One can add or remove entries from this table using the `+` and `-` qualifiers. Each entry in the table indicates the application and the priority that will be used for that application. Its value is of the form:

id/selector/priority

The meaning of the *id* is determined by the *selector* field. The *selector* field can be any one of the following:

- 1 — *id* indicates an Ethertype (an L2 protocol), therefore *id*'s value should be greater than 1536
- 2 — *id* indicates a port number over TCP or SCTP
- 3 — *id* indicates a port number over UDP or DCCP
- 4 — *id* indicates a port number over TCP, SCTP, UDP, or DCCP

The *priority* indicates the priority value (0-7) that will be used for given application.

Authorizations

The following subcommands require solaris.network.lldp authorization:

- `set-agentprop`
- `reset-agentprop`
- `set-tlvprop`
- `reset-tlvprop`
- `set-agenttlvprop`
- `reset-agenttlvprop`

The various `show-*` subcommands do not need any authorization.

使用例

例 1 Enabling LLDP Protocol on an LLDP Agent

The following command enables the LLDP protocol on an LLDP agent for both transmission and reception of LLDPDUs.

```
# lldpadm set-agentprop -p mode=both net0
```

例 2 Disabling LLDP Protocol on an LLDP Agent

The following command disables the LLDP protocol on an LLDP agent.

```
# lldpadm set-agentprop -p mode=disable net0
```

例 3 Configuring TLVs

The following command configures transmission of the Port Description and System Name TLV.

```
# lldpadm set-agentprop -p basic-tlv=portdesc,sysname net0
```

The following command configures transmission of a VLAN Name TLV and a Link Aggregation TLV.

```
# lldpadm set-agentprop -p dot1-tlv=vlanname net0
# lldpadm set-agentprop -p dot1-tlv+=linkaggr net0
```

The following command configures transmission of all `dot3-tlv`s.

```
# lldpadm set-agentprop -p dot3-tlv=all net0
```

All the above `lldpadm` invocations can be combined into the following, single invocation.

例 3 Configuring TLVs (続き)

```
# lldpadm set-agentprop -p basic-tlv=portdesc,sysname,\
dot1-tlv=vlanname,linkaggr,dot3-tlv=all net0
```

例 4 Disabling Transmission

The following command disables the transmission of all `dot1-tlv`s out of an LLDP agent.

```
# lldpadm set-agentprop -p dot1-tlv=none net0
```

The following command is equivalent to the preceding.

```
# lldpadm reset-agentprop -p dot1-tlv net0
```

例 5 Configuring Enabled Capabilities

The following command configures the enabled capabilities on a system.

```
# lldpadm set-tlvprop -p enabled=router syscapab
```

With this configuration, when an LLDP agent is enabled for advertising a System Capabilities TLV, the adjacent neighbors would learn of the local system's capabilities.

例 6 Configuring a Management Address for Subsequent Advertising

The following command configures the management address that will be advertised by means of the Management Address TLV.

```
# lldpadm set-tlvprop -p ipaddr=192.168.1.2 mgmtaddr
```

Note that this address would be identified as an address associated with the local LLDP agent that will be used to reach higher layer entities to assist discovery by network management.

例 7 Configuring an Application TLV

The following sequence of commands configures the application TLV to advertise the priority that will be used by FCoE.

```
# lldpadm set-agenttlvprop -p apt=8906/1/4 -a net0 appln
# lldpadm show-agenttlvprop -a net0 appln
AGENT    TLVNAME PROPERTY  PERM VALUE        DEFAULT        POSSIBLE
net0     appln   apt         rw 8906/1/4     --             --
```

例 8 Show Local Information Advertised by LLDP Agent

The following commands show, respectively, brief and detailed local information advertised by an LLDP agent.

Brief information:

```
# lldpadm show-agent -l net0
AGENT               CHASSISID           PORTID
```

例 8 Show Local Information Advertised by LLDP Agent (続き)

```
net0                004bb87f            00:14:4f:01:77:5d
```

Detailed information:

```
# lldpadm show-agent -lv net0
                     Agent: net0
        Chassis ID Subtype: Local(7)
                Chassis ID: 004bb87f
           Port ID Subtype: MacAddress(3)
                   Port ID: 00:14:4f:01:77:5d
          Port Description: net0
              Time to Live: 81 (seconds)
               System Name: hosta.example.com
        System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
    Supported Capabilities: bridge, router, station
      Enabled Capabilities: router
        Management Address: 192.168.1.2
        Maximum Frame Size: 3000
              Port VLAN ID: --
              VLAN Name/ID: vlan25/25
      VNIC PortID/VLAN ID : --
   Aggregation Information: Capable, Not Aggregated
               PFC Willing: --
                   PFC Cap: --
                   PFC MBC: --
                PFC Enable: --
Application(s)(ID/Sel/Pri): --
  Next Packet Transmission: 18 (seconds)
```

例 9 Show Remote Information about Adjacent Devices for an LLDP Agent

The following commands show, respectively, brief and detailed remote information about adjacent devices for a given LLDP agent.

Brief information:

```
# lldpadm show-agent -r net0
AGENT      SYSNAME      CHASSISID       PORTID
net0       hostb        0083b390        00:14:4f:01:59:ab
```

Detailed information:

```
# lldpadm show-agent -rv net0
                     Agent: net0
        Chassis ID Subtype: Local(7)
                Chassis ID: 0083b390
           Port ID Subtype: MacAddress(3)
                   Port ID: 00:14:4f:01:59:ab
```

例 9 Show Remote Information about Adjacent Devices for an LLDP Agent (続き)

```
          Port Description: net0
              Time to Live: 121 (seconds)
               System Name: hostb.example.com
        System Description: SunOS 5.11 dcb-clone-x-01-19-11 i86pc
    Supported Capabilities: bridge, router, station
      Enabled Capabilities: router
        Management Address: 192.168.1.3
        Maximum Frame Size: 3000
              Port VLAN ID: --
              VLAN Name/ID: vlan25/25
      VNIC PortID/VLAN ID : 02:08:20:72:71:31
   Aggregation Information: Capable, Not Aggregated
               PFC Willing: --
                   PFC Cap: --
                   PFC MBC: --
                PFC Enable: --
Application(s)(ID/Sel/Pri): --
   Information Valid Until: 117 (seconds)
```

例 10 Show LLDP Agent Statistics

The following command displays LLDP agent statistics.

```
# lldpadm show-agent -s net0
AGENT IFRAMES IERR IDISCARD OFRAMES OLENERR TLVDISCARD TLVUNRECOG AGEOUT
net0       44    0        0      57       0          0          0      0
```

属性

See attributes(5) for descriptions of the following attributes:

/sbin

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/lldp
Interface Stability	Committed

関連項目

hostid(1), read(1), uname(1), dladm(1M), lldpd(1M), svcadm(1M), attributes(5)

IEEE Std 802.1AB-2009, IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery

IEEE Draft 802.1Qbb, Virtual Bridged Local Area Networks - Amendment : Priority-based Flow Control

IEEE Draft 802.1Qaz, Virtual Bridged Local Area Networks - Amendment XX: Enhanced Transmission Selection for Bandwidth Sharing between Traffic Classes

注意事項

The lldpd(1M) daemon that implements the LLDP protocol must be first enabled before using the lldpadm command. The lldpd daemon is controlled through the service management facility (SMF) service instance:

```
svc:/network/lldp:default
```

Use svcadm(1M) to enable this service.

名前

lldpd – Link Layer Discovery Protocol daemon

形式

`/usr/lib/lldpd`

`svc:/network/lldp:default`

機能説明

`lldpd` is a system daemon that implements the Link Layer Discovery Protocol as specified in IEEE Std. 802.1AB. The daemon also manages LLDP agents on physical datalinks. Please see `lldpadm(1M)` for more information on administering LLDP on LLDP agents.

The `lldpd` daemon is controlled through the service management facility (SMF) service instance:

`svc:/network/lldp:default`

This means, `svcadm(1M)` must be used to start, stop, restart, and refresh this daemon. This daemon is disabled by default. To enable LLDP on any of the physical datalinks, this daemon should be running.

The `lldpd` daemon is a project private interface and has no user-accessible options.

使用例

例 1 Enabling the Daemon

The following command enables the `lldpd` daemon.

```
# svcadm enable network/lldp
# svcs network/lldp
STATE          STIME    FMRI
online         Jan_26   svc:/network/lldp:default
```

例 2 Disabling the Daemon

The following command disables the `lldpd` daemon.

```
# svcadm disable network/lldp
# svcs network/lldp
STATE          STIME    FMRI
disabled       5:11:35  svc:/network/lldp:default
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/lldp
Interface Stability	Project Private

関連項目

`lldpadm(1M)`, `svcadm(1M)`, `attributes(5)`

IEEE Std 802.1AB-2009, IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery

名前

lms – アプリケーションに Intel AMT (Active Management Technology) へのアクセスを許可する

形式

```
lms
```

機能説明

lms (Local Manageability Service、LMS) を使用すると、アプリケーションがローカル HECI インタフェースを使用して Intel AMT (Intel Active Management Technology) ME (Management Engine) にアクセス可能になります。LMS は HECI ドライバに依存しています。

lms を使用するには、Intel AMT Manageability Interface ドライバがインストール済みであり、かつ ME に接続済みであることを確認してください。

lms は、デーモンとして実行するように設計されています。このサービスからのメッセージは、syslog に送信されます。LMS メッセージには、「LMS」がソースであることを示すマークが付けられます。

SMF を使用して LMS サービスを有効にする方法については、「使用例」を参照してください。

使用例

例 1　SMF を使用して LMS サービスを有効にする

次の例では、SMF を使用して LMS サービスを有効にします。

```
svcadm enable network/lms
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	x86
使用条件	system/management/intel-amt
インタフェースの安定性	流動的

関連項目

attributes(5), smf(5)

名前

localectr – customize and build new locales

形式

```
/usr/bin/localectr

/usr/bin/localectr -l locale1,locale2... [-d destination_path]
     [-c compiler_path] [-r 0 | 1 | 2] [-i pkginfo_template]
     [-p pkgname_prefix]

/usr/bin/localectr -h

/usr/bin/localectr -q

/usr/bin/localectr -V
```

機能説明

The `localectr` utility allows new locales to be customized and built. The output of `localectr` is an installable package containing the compiled shared object binary which contains the locale data information as well as a number of other supporting files that are required to have a fully working locale on the system.

Once generated, the package can be added to the system by using the `pkgadd(1M)` command and removed with `pkgrm(1M)`.

Depending on your default system login, you might have to reset your user environment after you add a locale. If `dtlogin(1X)` (for the CDE user environment) is the default system login, you need to restart `dtlogin`. No action is required if `gdm(1)` (for the Gnome user environment) is the default login.

There are two interfaces to `localectr`, command line (CLI) and graphical user interface (GUI). To customize the locale data, you must use the GUI. To create locales with standard locale data according to Unicode's Common Locale Data Repository (CLDR), the CLI is sufficient.

With the CLI it is also possible to generate several locales in a single step, with a separate package being generated for each locale. With the GUI, a single locale is processed at a time.

To launch the GUI, run the `localectr` command with no options. To run from the CLI, use the appropriate options as described below.

`localectr` uses the `localedef(1)` utility to build the locale data binary shared object. Therefore, access to a C compiler is required in order to run `localectr` successfully, as this is also required by `localedef`.

`localectr` is mainly concerned with locale data. However, in order to create a fully working locale on the system with `localectr`, many features, such as fonts, translations and input methods, are also required. Depending on what locales are already installed on the system, the relevant features might or might not be present on the system. `localectr` bundles locale data for the latest set of locales available in Unicode's CLDR. A user can also create a locale not available in CLDR by supplying her own data in the `localectr` GUI.

オプション

The following options are supported:

`-c` *compiler_path*
Specify the path to the C compiler that is used to compile the locale data into a shared object binary. Not required if the compiler is already in the user's `PATH`.

`-d` *destination_path*
Specify the path to the directory where the created package is to be stored.

`-h`
Displays the usage message.

`-i` *template_file*
The full path to user defined `pkginfo(4)` template file.

`-l` *locale1,locale2...*
Specify a comma-separated list of locale(s) to generate. Locale names are in the form: *locale.codeset@variant*, where *codeset* and *variant* are optional. The default and only allowed *codeset* is `UTF-8`. The default *variant* is `localectr`.

`-p`
The package name prefix

`-q`
Queries `localectr` for a complete list of locales for which locale data is defined in `localectr`. When `localectr` is run from the CLI, the locale(s) specified with the `-l` option must be on this list in order for an installable locale package to be generated.

`-r`
Specify the range of Unicode characters for which locale data rules in the `LC_CTYPE` and `LC_COLLATE` categories are to be generated. There are three valid options:

`0`
Locale data rules are restricted to the exemplar or to commonly used characters of the locale in question.

`1`
Locale data rules are restricted to the Unicode plane 0 characters, whose codepoints fall in the range `u0000-uFFFF`.

`2`
Locale data rules are generated for all codepoints defined in the latest version of Unicode that is supported by the system on which `localectr` is being run.

`-V`
Shows the version of this software.

使用例

例 1 Launching the GUI

The following example launches the `localectr` GUI.

```
example% localectr
```

例 2 Generating Locale for Afrikaans (South Africa) with Default Locale Data

The following example generates a package in the specified destination directory, which can be used to install the Afrikaans (af_ZA.UTF-8) locale on the system. The package name is composed of a prefix followed by the hyphen separated ISO-639 language code, the ISO-3166 country code, the locale encoding and an optional user-defined tag. The resulting package can then be added to the system using pkgadd(1M).

```
example% localectr -l af_ZA -d /tmp
```

例 3 Generating Several South Africa Locales with Full Unicode Range of Characters

The following example generates an installable package for each of the specified locales.

```
example% localectr -l af_ZA,en_ZA,xh_ZA,zu_ZA -d /tmp -r 2
```

例 4 Generating the Irish Locale with a User-Specified Tag

The following example will generate a locale whose full name is ga_IE.UTF-8@mycompanyname.

```
example% localectr -l ga_IE@mycompanyname -d /tmp
```

終了ステータス

The following exit codes are returned:

0
Successful completion

>0
An error occurred.

ファイル

/usr/bin/localectr
Wrapper script that launches locale creator.

/usr/lib/localectr
Jar files, scripts, and locale data repository needed to run the application.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWlocalecreator
Interface Stability	See below.

Interface stability is Committed for command-line options and is Uncommitted for other interfaces.

関連項目

locale(1), localedef(1), pkgadd(1M), pkgrm(1M), pkginfo(4), attributes(5)

dtlogin(1X), gdm(1)(these are not SunOS man pages)

International Language Environments Guide

名前

locator – location indicator control

形式

```
/usr/sbin/locator [-f | -n]
```

機能説明

The locator command sets or queries the state of the system locator if such a device exists.

Without options, the locator command reports the current state of the system.

The privileges required to use this command are hardware dependent. Typically, only the super user can get or set a locator.

オプション

The following options are supported:

`-f` Turns the locator off.

`-n` Turns the locator on.

使用例

例 1 Using the locator Command on a Platform Which Has a System Locator LED

When issued on a platform which has a system locator LED, the following command turns the locator on:

```
# locator -n
# locator
The 'system' locator is on
```

例 2 Using the locator Command on a Platform Which Does Not Have a System Locator LED

When issued on a platform which does not have a system locator LED, the following command attempts to turn the locator on. The command returns an error message.

```
# locator -n
'system' locator not found
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Invalid command line input.

`2` The requested operation failed.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/picl

関連項目

`attributes(5)`

名前

lockd – network lock daemon

形式

```
/usr/lib/nfs/lockd [-g graceperiod] [-l listen_min_backlog]
      [-t timeout] [nthreads]
```

機能説明

The `lockd` utility is part of the NFS lock manager, which supports record locking operations on NFS files. See `fcntl(2)` and `lockf(3C)`. The lock manager provides the following two functions:

- It forwards `fcntl(2)` locking requests for NFS mounted file systems to the lock manager on the NFS server.
- It generates local file locking operations in response to requests forwarded from lock managers running on NFS client machines.

State information kept by the lock manager about these locking requests can be lost if the `lockd` is killed or the operating system is rebooted. Some of this information can be recovered as follows. When the server lock manager restarts, it waits for a grace period for all client-site lock managers to submit reclaim requests. Client-site lock managers, on the other hand, are notified by the status monitor daemon, `statd(1M)`, of the restart and promptly resubmit previously granted lock requests. If the lock daemon fails to secure a previously granted lock at the server site, then it sends `SIGLOST` to a process.

Administrators can make changes to the startup parameters for `lockd` by logging in as root and using the `sharectl(1M)` command.

SMF Management

The `lockd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/nfs/nlockmgr
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

If it is disabled, it will be enabled by `mount_nfs(1M)`, `share_nfs(1M)`, and `automountd(1M)` unless its `application/auto_enable` property is set to `false`.

The `sharectl(1M)` command is used to manipulate the startup SMF parameters for `lockd`. Currently supported parameters are as follows:

`lockd_listen_backlog=`*num*
: Set connection queue length for `lockd` over a connection-oriented transport. The default and minimum value is `32`. Equivalent to `-l` option.

`lockd_servers=`*num*
: Maximum number of concurrent `lockd` requests. The default is `1024`. Equivalent to the *nthreads* operand.

`lockd_retransmit_timeout=`*num*
: Retransmit timeout, in seconds, before `lockd` retries. The default is `5`. Equivalent to `-t` option.

`grace_period=`*num*
: Grace period, in seconds, that all clients (both NLM and NFSv4) have to reclaim locks after a server reboot. This parameter also controls the NFSv4 lease interval. The default is `90`. Equivalent to `-g` option.

See `EXAMPLES`, below.

オプション

The following options are supported:

`-g` *graceperiod*
: Deprecated in favor of `grace_period`. Specify the number of seconds that all clients (both NLM and NFSv4) have to reclaim locks after the server reboots. It also controls the NFSv4 lease interval. This option is equivalent to the `grace_period` property described above.

`-l` *listen_min_backlog*
: Specify the listener backlog (*listen_min_backlog*). *listen_min_backlog* is the number connect requests that are queued and waiting to be processed before new connect requests start to get dropped. Equivalent of the `lockd_listen_backlog` property described above.

`-t` *timeout*
: Specify the number of seconds to wait before retransmitting a lock request to the remote server. The default value is 5 seconds. Equivalent of the `lockd_retransmit_timeout` property described above.

オペランド

nthreads
: Specify the maximum number of concurrent threads that the server can handle. This concurrency is achieved by up to *nthreads* threads created as needed in the kernel. *nthreads* should be based on the load expected on this server. If *nthreads* is not specified, the maximum number of concurrent threads will default to 1024. Equivalent of the `lockd_servers` property described above.

使用例

例 1　Setting a lockd Property

The following command sets `lockd_listen_backlog` to a new value:

```
# sharectl set -p lockd_listen_backlog=40 nfs
```

The `lockd_listen_backlog` and other `lockd` properties are described under `NOTES`, below.

例 2　Getting a lockd Property Value

The following command retrieves the value of the `lockd_listen_backlog` property.

```
% sharectl get -p lockd_listen_backlog nfs
lockd_listen_backlog=40
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目

svcs(1), automountd(1M), clear_locks(1M), mount_nfs(1M), share(1M), share_nfs(1M), sharectl(1M), statd(1M), svcadm(1M), fcntl(2), lockf(3C), attributes(5), smf(5)

注意事項

The lockd daemon might not exist in a future release of Solaris.

名前

lockfs – change or report file system locks

形式

`/usr/sbin/lockfs [-adefhnuw] [-c string] [file-system]...`

機能説明

`lockfs` is used to change and report the status of file system locks. `lockfs` reports the lock status and unlocks the file systems that were improperly left locked.

Using `lockfs` to lock a file system is discouraged because this requires extensive knowledge of SunOS internals to be used effectively and correctly.

When invoked with no arguments, `lockfs` lists the UFS file systems that are locked. If *file-system* is not specified, and `-a` is specified, `lockfs` is run on all mounted, UFS type file systems.

オプション

The options are mutually exclusive: `wndheuf`. If you do specify more than one of these options on a `lockfs` command line, the utility does not protest and invokes only the last option specified. In particular, you cannot specify a flush (`-f`) and a lock (for example, `-w`) on the same command line. However, all locking operations implicitly perform a flush, so the `-f` is superfluous when specifying a lock.

You must be super-user to use any of the following options, with the exception of `-a`, `-f` and `-v`.

The following options are supported.

`-a`
: Apply command to all mounted, UFS type file systems. *file-system* is ignored when `-a` is specified.

`-c` *string*
: Accept a string that is passed as the comment field. The `-c` only takes affect when the lock is being set using the `-d`, `-h`, `-n`, `-u`, or `-w` options.

`-d`
: Delete-lock (`dlock`) the specified *file-system*. dlock suspends access that could remove directory entries.

`-e`
: Error-lock (`elock`) the specified *file-system*. elock blocks all local access to the locked file system and returns `EWOULDBLOCK` on all remote access. File systems are elocked by UFS on detection of internal inconsistency. They may only be unlocked after successful repair by `fsck`, which is usually done automatically (see `mount_ufs(1M)`). elocked file systems can be unmounted.

`-f`
: Force a synchronous flush of all data that is dirty at the time `fsflush` is run to its backing store for the named file system (or for all file systems.)

It is a more reliable method than using sync(1M) because it does not return until all possible data has been pushed. In the case of UFS filesystems with logging enabled, the log is also rolled before returning. Additional data can be modified by the time `fsflush` exits, so using one of the locking options is more likely to be of general use.

`-h`
Hard-lock (`hlock`) the specified *file-system*. hlock returns an error on every access to the locked file system, and cannot be unlocked. hlocked file systems can be unmounted.

`-n`
Name-lock (`nlock`) the specified *file-system*. nlock suspends accesses that could change or remove existing directories entries.

`-u`
Unlock (`ulock`) the specified *file-system*. ulock awakens suspended accesses.

`-v`
Enable verbose output.

`-w`
Write-lock (`wlock`) the specified *file-system*. wlock suspends writes that would modify the file system. Access times are not kept while a file system is write-locked.

オペランド

The following operands are supported.

file-system
A list of path names separated by whitespace. Note that *file-system* can be a directory rather than the specific name of a file system, such as `/` or `/usr`. For example, if you specify `/export/home` as an argument to a `lockfs` command and `/export/home` is mounted on the root (`/`) file system, the `lockfs` command will take effect on the root file system.

使用法

See largefile(5) for the description of the behavior of `lockfs` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

使用例

例 1 Using lockfs -a

In the following examples, *filesystem* is the pathname of the mounted-on directory (mount point). `Locktype` is one of "`write`," "*name*," "*delete*," "*hard*," or "*unlock*". When enclosed in parenthesis, the lock is being set. `Comment` is a string set by the process that last issued a lock command.

The following example shows the `lockfs` output when only the `-a` option is specified.

```
example#  /usr/sbin/lockfs -a
```

Filesystem	Locktype	Comment
/	unlock	

例 1 Using lockfs -a (続き)

/var	unlock

```
example#
```

例 2 Using lockfs -w

The following example shows the `lockfs` output when the `-w` option is used to write lock the `/var` file system and the comment string is set using the `-c` option. The `-a` option is then specified on a separate command line.

```
example#  /usr/sbin/lockfs -w -c "lockfs: write lock example" /var
example#  /usr/sbin/lockfs -a
```

Filesystem	Locktype	Comment
/	unlock	
/var	write	lockfs: write lock example

```
example#
```

例 3 Using lockfs -u

The following example shows the `lockfs` output when the `-u` option is used to unlock the `/var` file system and the comment string is set using the `-c` option.

```
example#  /usr/sbin/lockfs -uc "lockfs: unlock example" /var
example#  /usr/sbin/lockfs /var
```

Filesystem	Locktype	Comment
/var	unlock	lockfs: unlock example

```
example#
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`kill(1)`, `mount_ufs(1M)`, `sync(1M)`, `attributes(5)`, `largefile(5)`, `ufs(7FS)`,

『Oracle Solaris の管理: 一般的なタスク』

診断

file system: `Not owner`
You must be root to use this command.

file system :`Deadlock condition detected/avoided`
A file is enabled for accounting or swapping, on *file system*.

file system: `Device busy`
Another process is setting the lock on *file system*.

名前

lockstat – report kernel lock and profiling statistics

形式

```
lockstat [-ACEHI] [-e event_list] [-i rate]
     [-b | -t | -h | -s depth] [-n nrecords]
     [-l lock [, size]] [-d duration]
     [-f function [, size]] [-T] [-ckgwWRpP] [-D count]
     [-o filename] [-x opt [=val]] command [args]
```

機能説明

The `lockstat` utility gathers and displays kernel locking and profiling statistics. `lockstat` allows you to specify which events to watch (for example, spin on adaptive mutex, block on read access to rwlock due to waiting writers, and so forth) how much data to gather for each event, and how to display the data. By default, `lockstat` monitors all lock contention events, gathers frequency and timing data about those events, and displays the data in decreasing frequency order, so that the most common events appear first.

`lockstat` gathers data until the specified command completes. For example, to gather statistics for a fixed-time interval, use `sleep(1)` as the command, as follows:

```
example# lockstat sleep 5
```

When the `-I` option is specified, `lockstat` establishes a per-processor high-level periodic interrupt source to gather profiling data. The interrupt handler simply generates a `lockstat` event whose caller is the interrupted PC (program counter). The profiling event is just like any other `lockstat` event, so all of the normal `lockstat` options are applicable.

`lockstat` relies on DTrace to modify the running kernel's text to intercept events of interest. This imposes a small but measurable overhead on all system activity, so access to `lockstat` is restricted to super-user by default. The system administrator can permit other users to use `lockstat` by granting them additional DTrace privileges. Refer to the Solaris Dynamic Tracing Guide for more information about DTrace security features.

オプション

The following options are supported:

Event Selection

If no event selection options are specified, the default is `-C`.

`-A`
: Watch all lock events. `-A` is equivalent to `-CH`.

`-C`
: Watch contention events.

`-E`
: Watch error events.

`-e` *event_list*
: Only watch the specified events. *event list* is a comma-separated list of events or ranges of events such as 1,4-7,35. Run `lockstat` with no arguments to get a brief description of all events.

`-H`
Watch hold events.

`-I`
Watch profiling interrupt events.

`-i` *rate*
Interrupt rate (per second) for `-I`. The default is 97 Hz, so that profiling doesn't run in lockstep with the clock interrupt (which runs at 100 Hz).

Data Gathering

`-x` *arg*[=*val*]
Enable or modify a DTrace runtime option or D compiler option. The list of options is found in `dtrace(1M)`. Boolean options are enabled by specifying their name. Options with values are set by separating the option name and value with an equals sign (=).

Data Gathering (Mutually Exclusive)

`-b`
Basic statistics: lock, caller, number of events.

`-h`
Histogram: Timing plus time-distribution histograms.

`-s` *depth*
Stack trace: Histogram plus stack traces up to *depth* frames deep.

`-t`
Timing: Basic plus timing for all events [default].

Data Filtering

`-d` *duration*
Only watch events longer than *duration*.

`-f` *func[,size]*
Only watch events generated by *func*, which can be specified as a symbolic name or hex address. *size* defaults to the ELF symbol size if available, or `1` if not.

`-l` *lock[,size]*
Only watch *lock*, which can be specified as a symbolic name or hex address. `size` defaults to the ELF symbol size or `1` if the symbol size is not available.

`-n` *nrecords*
Maximum number of data records.

`-T`
Trace (rather than sample) events [off by default].

Data Reporting

`-c`
Coalesce lock data for lock arrays (for example, `pse_mutex[]`).

`-D` *count*
Only display the top *count* events of each type.

`-g`
Show total events generated by function. For example, if `foo()` calls `bar()` in a loop, the work done by `bar()` counts as work generated by `foo()` (along with any work done by `foo()` itself). The `-g` option works by counting the total number of stack frames in which each function appears. This implies two things: (1) the data reported by `-g` can be misleading if the stack traces are not deep enough, and (2) functions that are called recursively might show greater than 100% activity. In light of issue (1), the default data gathering mode when using `-g` is `-s 50`.

`-k`
Coalesce PCs within functions.

`-o` *filename*
Direct output to *filename*.

`-P`
Sort data by (*count* * *time*) product.

`-p`
Parsable output format.

`-R`
Display rates (events per second) rather than counts.

`-W`
Whichever: distinguish events only by caller, not by lock.

`-w`
Wherever: distinguish events only by lock, not by caller.

Display Formats

The following headers appear over various columns of data.

`Count` or `ops/s`
Number of times this event occurred, or the rate (times per second) if `-R` was specified.

`indv`
Percentage of all events represented by this individual event.

`genr`
Percentage of all events generated by this function.

`cuml`
Cumulative percentage; a running total of the individuals.

`rcnt`
Average reference count. This will always be `1` for exclusive locks (mutexes, spin locks, rwlocks held as writer) but can be greater than `1` for shared locks (rwlocks held as reader).

`nsec`
Average duration of the events in nanoseconds, as appropriate for the event. For the profiling event, duration means interrupt latency.

`Lock`
: Address of the lock; displayed symbolically if possible.

`CPU+PIL`
: CPU plus processor interrupt level (PIL). For example, if CPU 4 is interrupted while at PIL 6, this will be reported as `cpu[4]+6`.

`Caller`
: Address of the caller; displayed symbolically if possible.

使用例

例 1 Measuring Kernel Lock Contention

```
example# lockstat sleep 5
Adaptive mutex spin: 2210 events in 5.055 seconds (437 events/sec)

Count indv cuml rcnt     nsec Lock                 Caller
-------------------------------------------------------------------------
  269  12%  12% 1.00     2160 service_queue        background+0xdc
  249  11%  23% 1.00       86 service_queue        qenable_locked+0x64
  228  10%  34% 1.00      131 service_queue        background+0x15c
   68   3%  37% 1.00       79 0x30000024070        untimeout+0x1c
   59   3%  40% 1.00      384 0x300066fa8e0        background+0xb0
   43   2%  41% 1.00       30 rqcred_lock          svc_getreq+0x3c
   42   2%  43% 1.00      341 0x30006834eb8        background+0xb0
   41   2%  45% 1.00      135 0x30000021058        untimeout+0x1c
   40   2%  47% 1.00       39 rqcred_lock          svc_getreq+0x260
   37   2%  49% 1.00     2372 0x300068e83d0        hmestart+0x1c4
   36   2%  50% 1.00       77 0x30000021058        timeout_common+0x4
   36   2%  52% 1.00      354 0x300066fa120        background+0xb0
   32   1%  53% 1.00       97 0x30000024070        timeout_common+0x4
   31   1%  55% 1.00     2923 0x300069883d0        hmestart+0x1c4
   29   1%  56% 1.00      366 0x300066fb290        background+0xb0
   28   1%  57% 1.00      117 0x3000001e040        untimeout+0x1c
   25   1%  59% 1.00       93 0x3000001e040        timeout_common+0x4
   22   1%  60% 1.00       25 0x30005161110        sync_stream_buf+0xdc
   21   1%  60% 1.00      291 0x30006834eb8        putq+0xa4
   19   1%  61% 1.00       43 0x3000515dcb0        mdf_alloc+0xc
   18   1%  62% 1.00      456 0x30006834eb8        qenable+0x8
   18   1%  63% 1.00       61 service_queue        queuerun+0x168
   17   1%  64% 1.00      268 0x30005418ee8        vmem_free+0x3c
[...]

R/W reader blocked by writer: 76 events in 5.055 seconds (15 events/sec)

Count indv cuml rcnt     nsec Lock                 Caller
-------------------------------------------------------------------------
   23  30%  30% 1.00 22590137 0x300098ba358        ufs_dirlook+0xd0
   17  22%  53% 1.00  5820995 0x3000ad815e8        find_bp+0x10
   13  17%  70% 1.00  2639918 0x300098ba360        ufs_iget+0x198
    4   5%  75% 1.00  3193015 0x300098ba360        ufs_getattr+0x54
```

例 1　Measuring Kernel Lock Contention　　*(続き)*

```
    3   4%  79% 1.00  7953418 0x3000ad817c0       find_bp+0x10
    3   4%  83% 1.00   935211 0x3000ad815e8       find_read_lof+0x14
    2   3%  86% 1.00 16357310 0x300073a4720       find_bp+0x10
    2   3%  88% 1.00  2072433 0x300073a4720       find_read_lof+0x14
    2   3%  91% 1.00  1606153 0x300073a4370       find_bp+0x10
    1   1%  92% 1.00  2656909 0x300107e7400       ufs_iget+0x198
[...]
```

例 2　Measuring Hold Times

```
example# lockstat -H -D 10 sleep 1
Adaptive mutex spin: 513 events

Count indv cuml rcnt     nsec Lock                 Caller
-------------------------------------------------------------------------
  480   5%   5% 1.00     1136 0x300007718e8        putnext+0x40
  286   3%   9% 1.00      666 0x3000077b430        getf+0xd8
  271   3%  12% 1.00      537 0x3000077b430        msgio32+0x2fc
  270   3%  15% 1.00     3670 0x300007718e8        strgetmsg+0x3d4
  270   3%  18% 1.00     1016 0x300007c38b0        getq_noenab+0x200
  264   3%  20% 1.00     1649 0x300007718e8        strgetmsg+0xa70
  216   2%  23% 1.00     6251 tcp_mi_lock          tcp_snmp_get+0xfc
  206   2%  25% 1.00      602 thread_free_lock     clock+0x250
  138   2%  27% 1.00      485 0x300007c3998        putnext+0xb8
  138   2%  28% 1.00     3706 0x300007718e8        strrput+0x5b8
-------------------------------------------------------------------------
[...]
```

例 3　Measuring Hold Times for Stack Traces Containing a Specific Function

```
example# lockstat -H -f tcp_rput_data -s 50 -D 10 sleep 1
Adaptive mutex spin: 11 events in 1.023 seconds (11
events/sec)

-------------------------------------------------------------------------
Count indv cuml rcnt     nsec Lock                    Caller
    9  82%  82% 1.00     2540 0x30000031380           tcp_rput_data+0x2b90

      nsec ------ Time Distribution ------ count     Stack
       256 |@@@@@@@@@@@@@@@@@               5         tcp_rput_data+0x2b90
       512 |@@@@@@                          2         putnext+0x78
      1024 |@@@                             1         ip_rput+0xec4
      2048 |                                0         _c_putnext+0x148
      4096 |                                0         hmeread+0x31c
      8192 |                                0         hmeintr+0x36c
     16384 |@@@                             1
sbus_intr_wrapper+0x30
[...]
```

例 3　Measuring Hold Times for Stack Traces Containing a Specific Function　(続き)

```
Count indv cuml rcnt     nsec Lock                   Caller
    1   9%  91% 1.00     1036 0x30000055380          freemsg+0x44

      nsec ------ Time Distribution ------ count     Stack
      1024 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 1       freemsg+0x44
                                                     tcp_rput_data+0x2fd0
                                                     putnext+0x78
                                                     ip_rput+0xec4
                                                     _c_putnext+0x148
                                                     hmeread+0x31c
                                                     hmeintr+0x36c

sbus_intr_wrapper+0x30
-------------------------------------------------------------------------
[...]
```

例 4　Basic Kernel Profiling

For basic profiling, we don't care whether the profiling interrupt sampled `foo()+0x4c` or `foo()+0x78`; we care only that it sampled somewhere in `foo()`, so we use `-k`. The CPU and PIL aren't relevant to basic profiling because we are measuring the system as a whole, not a particular CPU or interrupt level, so we use `-W`.

```
example# lockstat -kIW -D 20 ./polltest
Profiling interrupt: 82 events in 0.424 seconds (194
events/sec)

Count indv cuml rcnt     nsec Hottest CPU+PIL     Caller
-------------------------------------------------------------------------
    8  10%  10% 1.00      698 cpu[1]              utl0
    6   7%  17% 1.00      299 cpu[0]              read
    5   6%  23% 1.00      124 cpu[1]              getf
    4   5%  28% 1.00      327 cpu[0]              fifo_read
    4   5%  33% 1.00      112 cpu[1]              poll
    4   5%  38% 1.00      212 cpu[1]              uiomove
    4   5%  43% 1.00      361 cpu[1]              mutex_tryenter
    3   4%  46% 1.00      682 cpu[0]              write
    3   4%  50% 1.00       89 cpu[0]              pcache_poll
    3   4%  54% 1.00      118 cpu[1]              set_active_fd
    3   4%  57% 1.00      105 cpu[0]              syscall_trap32
    3   4%  61% 1.00      640 cpu[1]              (usermode)
    2   2%  63% 1.00      127 cpu[1]              fifo_poll
    2   2%  66% 1.00      300 cpu[1]              fifo_write
    2   2%  68% 1.00      669 cpu[0]              releasef
    2   2%  71% 1.00      112 cpu[1]              bt_getlowbit
    2   2%  73% 1.00      247 cpu[1]              splx
```

例 4 Basic Kernel Profiling *(続き)*

```
    2   2%  76% 1.00      503 cpu[0]              mutex_enter
    2   2%  78% 1.00      467 cpu[0]+10           disp_lock_enter
    2   2%  80% 1.00      139 cpu[1]              default_copyin
-------------------------------------------------------------------------
[...]
```

例 5 Generated-load Profiling

In the example above, 5% of the samples were in `poll()`. This tells us how much time was spent inside `poll()` itself, but tells us nothing about how much work was *generated* by `poll()`; that is, how much time we spent in functions called by `poll()`. To determine that, we use the `-g` option. The example below shows that although `polltest` spends only 5% of its time in `poll()` itself, `poll()`-induced work accounts for 34% of the load.

Note that the functions that generate the profiling interrupt (`lockstat_intr()`, `cyclic_fire()`, and so forth) appear in every stack trace, and therefore are considered to have generated 100% of the load. This illustrates an important point: the generated load percentages do *not* add up to 100% because they are not independent. If 72% of all stack traces contain both `foo()` and `bar()`, then both `foo()` and `bar()` are 72% load generators.

```
example# lockstat -kgIW -D 20 ./polltest
Profiling interrupt: 80 events in 0.412 seconds (194 events/sec)

Count genr cuml rcnt     nsec Hottest CPU+PIL     Caller
-------------------------------------------------------------------------
   80 100% ---- 1.00      310 cpu[1]              lockstat_intr
   80 100% ---- 1.00      310 cpu[1]              cyclic_fire
   80 100% ---- 1.00      310 cpu[1]              cbe_level14
   80 100% ---- 1.00      310 cpu[1]              current_thread
   27  34% ---- 1.00      176 cpu[1]              poll
   20  25% ---- 1.00      221 cpu[0]              write
   19  24% ---- 1.00      249 cpu[1]              read
   17  21% ---- 1.00      232 cpu[0]              write32
   17  21% ---- 1.00      207 cpu[1]              pcache_poll
   14  18% ---- 1.00      319 cpu[0]              fifo_write
   13  16% ---- 1.00      214 cpu[1]              read32
   10  12% ---- 1.00      208 cpu[1]              fifo_read
   10  12% ---- 1.00      787 cpu[1]              utl0
    9  11% ---- 1.00      178 cpu[0]              pcacheset_resolve
    9  11% ---- 1.00      262 cpu[0]              uiomove
    7   9% ---- 1.00      506 cpu[1]              (usermode)
    5   6% ---- 1.00      195 cpu[1]              fifo_poll
    5   6% ---- 1.00      136 cpu[1]              syscall_trap32
    4   5% ---- 1.00      139 cpu[0]              releasef
    3   4% ---- 1.00      277 cpu[1]              polllock
-------------------------------------------------------------------------
[...]
```

例 6 Gathering Lock Contention and Profiling Data for a Specific Module

In this example we use the -f option not to specify a single function, but rather to specify the entire text space of the sbus module. We gather both lock contention and profiling statistics so that contention can be correlated with overall load on the module.

```
example# modinfo | grep sbus
 24 102a8b6f   b8b4  59   1  sbus (SBus (sysio) nexus driver)

example# lockstat -kICE -f 0x102a8b6f,0xb8b4 sleep 10
Adaptive mutex spin: 39 events in 10.042 seconds (4 events/sec)

Count indv cuml rcnt     nsec Lock               Caller
-------------------------------------------------------------------------
   15  38%  38% 1.00      206 0x30005160528      sync_stream_buf
    7  18%  56% 1.00       14 0x30005160d18      sync_stream_buf
    6  15%  72% 1.00       27 0x300060c3118      sync_stream_buf
    5  13%  85% 1.00       24 0x300060c3510      sync_stream_buf
    2   5%  90% 1.00       29 0x300060c2d20      sync_stream_buf
    2   5%  95% 1.00       24 0x30005161cf8      sync_stream_buf
    1   3%  97% 1.00       21 0x30005161110      sync_stream_buf
    1   3% 100% 1.00       23 0x30005160130      sync_stream_buf
[...]

Adaptive mutex block: 9 events in 10.042 seconds (1 events/sec)

Count indv cuml rcnt     nsec Lock               Caller
-------------------------------------------------------------------------
    4  44%  44% 1.00   156539 0x30005160528      sync_stream_buf
    2  22%  67% 1.00   763516 0x30005160d18      sync_stream_buf
    1  11%  78% 1.00   462130 0x300060c3510      sync_stream_buf
    1  11%  89% 1.00   288749 0x30005161110      sync_stream_buf
    1  11% 100% 1.00  1015374 0x30005160130      sync_stream_buf
[...]

Profiling interrupt: 229 events in 10.042 seconds (23 events/sec)

Count indv cuml rcnt     nsec Hottest CPU+PIL    Caller

-------------------------------------------------------------------------
   89  39%  39% 1.00      426 cpu[0]+6           sync_stream_buf
   64  28%  67% 1.00      398 cpu[0]+6           sbus_intr_wrapper
   23  10%  77% 1.00      324 cpu[0]+6           iommu_dvma_kaddr_load
   21   9%  86% 1.00      512 cpu[0]+6           iommu_tlb_flush
   14   6%  92% 1.00      342 cpu[0]+6           iommu_dvma_unload
   13   6%  98% 1.00      306 cpu[1]             iommu_dvma_sync
    5   2% 100% 1.00      389 cpu[1]             iommu_dma_bindhdl
-------------------------------------------------------------------------
[...]
```

例 7 Determining the Average PIL (processor interrupt level) for a CPU

```
example# lockstat -Iw -l cpu[3] ./testprog

Profiling interrupt: 14791 events in 152.463 seconds (97 events/sec)

Count indv cuml rcnt     nsec CPU+PIL            Hottest Caller

-----------------------------------------------------------------------
13641  92%  92% 1.00      253 cpu[3]             (usermode)
  579   4%  96% 1.00      325 cpu[3]+6           ip_ocsum+0xe8
  375   3%  99% 1.00      411 cpu[3]+10          splx
  154   1% 100% 1.00      527 cpu[3]+4           fas_intr_svc+0x80
   41   0% 100% 1.00      293 cpu[3]+13          send_mondo+0x18
    1   0% 100% 1.00      266 cpu[3]+12          zsa_rxint+0x400
-----------------------------------------------------------------------
[...]
```

例 8 Determining which Subsystem is Causing the System to be Busy

```
example# lockstat -s 10 -I sleep 20

Profiling interrupt: 4863 events in 47.375 seconds (103 events/sec)

Count indv cuml rcnt     nsec CPU+PIL         Caller

-----------------------------------------------------------------------
1929   40%  40% 0.00     3215 cpu[0]          usec_delay+0x78
  nsec ------ Time Distribution ------ count    Stack
  4096 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  1872    ata_wait+0x90
  8192 |                                27      acersb_get_intr_status+0x34
 16384 |                                29      ata_set_feature+0x124
 32768 |                                1       ata_disk_start+0x15c
                                                ata_hba_start+0xbc
                                                ghd_waitq_process_and \
                                                _mutex_hold+0x70
                                                ghd_waitq_process_and \
                                                _mutex_exit+0x4
                                                ghd_transport+0x12c
                                                ata_disk_tran_start+0x108
-----------------------------------------------------------------------
[...]
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/dtrace

関連項目

dtrace(1M), plockstat(1M), attributes(5), lockstat(7D), mutex(9F), rwlock(9F)

Solaris Dynamic Tracing Guide

注意事項

The profiling support provided by `lockstat -I` replaces the old (and undocumented) `/usr/bin/kgmon` and `/dev/profile`.

Tail-call elimination can affect call sites. For example, if `foo()+0x50` calls `bar()` and the last thing `bar()` does is call `mutex_exit()`, the compiler can arrange for `bar()` to branch to `mutex_exit()` with a return address of `foo()+0x58`. Thus, the `mutex_exit()` in `bar()` will appear as though it occurred at `foo()+0x58`.

The PC in the stack frame in which an interrupt occurs can be bogus because, between function calls, the compiler is free to use the return address register for local storage.

When using the `-I` and `-s` options together, the interrupted PC will usually not appear anywhere in the stack since the interrupt handler is entered asynchronously, not by a function call from that PC.

The `lockstat` technology is provided on an as-is basis. The format and content of `lockstat` output reflect the current Solaris kernel implementation and are therefore subject to change in future releases.

名前

lofiadm – lofi を使用してブロックデバイスとして使用可能なファイルを管理する

形式

```
/usr/sbin/lofiadm -a file [device]

/usr/sbin/lofiadm -c crypto_algorithm -a file [device]

/usr/sbin/lofiadm -c crypto_algorithm -k raw_key_file -a file [device]

/usr/sbin/lofiadm -c crypto_algorithm -T token_key -a file [device]

/usr/sbin/lofiadm -c crypto_algorithm -T token_key
     -k wrapped_key_file -a file [device]

/usr/sbin/lofiadm -c crypto_algorithm -e -a file [device]

/usr/sbin/lofiadm -C algorithm [-s segment_size] file

/usr/sbin/lofiadm -d file | device

/usr/sbin/lofiadm -U file

/usr/sbin/lofiadm [ file | device]
```

機能説明

lofiadm は lofi (ループバックファイルドライバ) を管理します。lofi を使用すると、ファイルをブロックデバイスに関連付けることができます。関連付けられたファイルは、ブロックデバイスからアクセスできます。これはファイルになんらかのファイルシステムのイメージ (CD-ROM イメージなど) が含まれている場合に役立ちます。ファイルシステムのマウント、検査、または修復用の標準のシステムユーティリティーでそのブロックデバイスを使用できるからです。fsck(1M) および mount(1M) を参照してください。

lofiadm を使用して、ファイルをループバックデバイスとして追加したり、そうした関連付けを削除したり、現在の関連付けに関する情報を表示したりします。

利用可能な lofi デバイスの数は、zone.max-lofi rctl により制限されます。これは、大域ゾーンで zonecfg(1M) を使用すると設定できます。zone.max-lofi については、resource_controls(5) を参照してください。

暗号化オプションと圧縮のオプションの両方をコマンド行に指定することはできません。また、暗号化されたファイルをあとで圧縮することも、圧縮されたファイルをあとで暗号化することもできません。

大域ゾーンでは、大域ゾーンデバイス、およびシステムのほかの非大域ゾーンにより所有されるすべてのデバイスの両方で lofiadm を使用できます。

オプション

次のオプションがサポートされています。

-a *file* [*device*]
: *file* をブロックデバイスとして追加します。

  device が指定されていない場合、使用可能なデバイスが 1 つ選択されます。

device が指定されている場合、`lofiadm` はそのデバイスの *file* への割り当てを試みます。*device* は使用可能でなければならず、そうでない場合は `lofiadm` は失敗します。デバイスを指定する機能は、特定の関連付けセットを再設定するスクリプトで使用するためのものです。

`-C` {*gzip* | *gzip-N* | *lzma*}
指定された圧縮アルゴリズムでファイルを圧縮します。

`gzip` 圧縮アルゴリズムでは、オープンソースの `gzip` コマンドと同じ圧縮が使用されます。`gzip` のレベルを値 `gzip-`*N* を使用して指定できます。この *N* は 6 (高速) または 9 (最大圧縮率) です。現在、数値を指定しない `gzip` は、`gzip-6` (`gzip` コマンドのデフォルトでもある) に相当します。

lzma は、LZMA (Lempel-Ziv-Markov) 圧縮アルゴリズムを表します。

圧縮済みのファイルに書き込むことも、圧縮済みのファイルを読み取り/書き込みでマウントすることもできません。

`-d` *file* | *device*
関連付けられたブロックデバイスがビジー状態ではない場合は、*file* または *device* 名で指定された関連付けを削除し、ブロックデバイスの割り当てを解除します。

`-s` *segment_size*
圧縮するファイルの分割に使用するセグメントサイズ。*segment_size* は整数で、512 の倍数です。

`-U` *file*
圧縮済みのファイルを圧縮解除します。

ファイルが暗号化されている場合は、次のオプションを使用します。

`-c` *crypto_algorithm*
暗号化アルゴリズムを選択します。アルゴリズムはディスクイメージに格納されていないため、暗号化が有効な場合はアルゴリズムを指定する必要があります。

`-e`、`-k`、`-T` のいずれも指定されていない場合、`lofiadm` は 8 文字以上のパスフレーズの入力を求めます。PKCS#5 PBKD2 を利用した対称暗号化鍵の取得には、パスフレーズが使用されます。

`-k` *raw_key_file* | *wrapped_key_file*
raw またはラップされた対称暗号化鍵へのパス。PKCS#11 オブジェクトも `-T` オプションを使って指定される場合、そのオブジェクトにより鍵がラップされます。`-T` が指定されていない場合、raw の鍵が使用されます。

`-T` *token_key*
暗号化または鍵ファイルのラップ解除に使用される PKCS#11 トークン内の鍵。

`-k` も指定されている場合、`-T` でラップを解除する鍵を特定します。この鍵は RSA 秘密鍵である必要があります。

`-e`
一時的な暗号化鍵を生成します。

オペランド

次のオペランドがサポートされています。

crypto_algorithm
`aes-128-cbc`、`aes-192-cbc`、`aes-256-cbc`、`des3-cbc`、`blowfish-cbc` のいずれか。

device
ブロックデバイス *device* と関連付けられたファイル名を表示します。

引数が指定されていない場合、現在の関連付けのリストを一覧表示します。ファイル名は有効な絶対パス名である必要があります。

ファイルが追加されると、root による読み取りまたは書き込みのためにファイルが開かれます。すべての制限が適用されます (NFS 上のルートアクセスの制限など)。ファイルは、関連付けが削除されるまで開かれたままになります。ブロックデバイスが使用されるまでファイルは実際にアクセスされないため、ブロックデバイスが読み取り専用で開かれた場合はファイルに書き込まれません。

現在のコンテキストでパスを解決できない場合 (非大域ゾーン内の NFS パスなど)、ファイル名が疑問符 (?) で表示されることがあります。

file
file に関連付けられているブロックデバイスを表示します。

raw_key_file
raw 対称暗号化鍵に使用する、適切な長さ (単位はビット) のファイルへのパス。

token_key
PKCS#11 トークンオブジェクト。書式は次のとおりです。

token_name:*manufacturer_id*:*serial_number*:*key_label*

鍵ラベル以外はすべて省略可能であり、空でもかまいません。たとえば、鍵ラベル `MylofiKey` のみを使ってトークンオブジェクトを指定するには、次のように記述します。

```
-T :::MylofiKey
```

wrapped_key_file
`-T` で指定された RSA 秘密鍵を使ってラップされた対称暗号化鍵を含むファイルへのパス。

使用例

例 1　既存の CD-ROM イメージのマウント

CD を作成する前に、Solaris がイメージを認識することを確認する必要があります。`lofi` を使用すればイメージをマウントでき、これが機能するかどうかを確認できます。

例 1　既存の CD-ROM イメージのマウント　　(続き)

この例では、インターネットからダウンロードした Red Hat 6.0 CD の既存の CD-ROM イメージ (sparc.iso) をマウントします。これはインターネットから入手した mkisofs ユーティリティーで作成されました。

次のように lofiadm を使用して、これにブロックデバイスを接続します。

```
# lofiadm -a /home/mike_s/RH6.0/sparc.iso
/dev/lofi/1
```

lofiadm はデバイスを選択し、デバイス名を標準出力に表示します。次のコマンドを実行して、lofiadm を再度実行できます。

```
# lofiadm
Block Device     File                          Options
/dev/lofi/1      /home/mike_s/RH6.0/sparc.iso   -
```

または、次のコマンドを実行して一方の名前を指定し、もう一方の名前を要求できます。

```
# lofiadm /dev/lofi/1
/home/mike_s/RH6.0/sparc.iso
```

mount コマンドを使用してイメージをマウントします。

```
# mount -F hsfs -o ro /dev/lofi/1 /mnt
```

Solaris がイメージを認識することを確認します。

```
# df -k /mnt
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/lofi/1           512418  512418       0   100%    /mnt
# ls /mnt
./            RedHat/       doc/          ls-lR         rr_moved/
../           TRANS.TBL     dosutils/     ls-lR.gz      sbin@
.buildlog     bin@          etc@          misc/         tmp/
COPYING       boot/         images/       mnt/          usr@
README        boot.cat*     kernels/      modules/
RPM-PGP-KEY   dev@          lib@          proc/
```

Solaris は CD-ROM イメージをマウントし、ファイル名を認識できます。イメージが正しく作成されたため、確信をもって CD-ROM を作成できます。

最終ステップとして、イメージのマウント解除および切り離しを行います。

```
# umount /mnt
# lofiadm -d /dev/lofi/1
# lofiadm
Block Device             File             Options
```

例 2　ファイル上の UFS ファイルシステムの作成

UFS ファイルシステムをファイル上に作成すると、特にテストスイートに新しいファイルシステムが必要な場合に便利です。テストスイートのためだけにディスクをパーティションに再分割することは手間がかかりますが、その必要はありません。lofi を使用するとファイルに newfs を実行できます。

ファイルを作成します。

```
# mkfile 35m /export/home/test
```

作成したファイルをブロックデバイスに接続します。newfs に必要な文字デバイスも取得するため、newfs は次のようになります。

```
# lofiadm -a /export/home/test
/dev/lofi/1
# newfs /dev/rlofi/1
newfs: construct a new file system /dev/rlofi/1: (y/n)? y
/dev/rlofi/1:   71638 sectors in 119 cylinders of 1 tracks, 602 sectors
        35.0MB in 8 cyl groups (16 c/g, 4.70MB/g, 2240 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 9664, 19296, 28928, 38560, 48192, 57824, 67456,
```

ufs はファイル全体を使用できない場合があります。ファイルシステムをマウントおよび使用します。

```
# mount /dev/lofi/1 /mnt
# df -k /mnt
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/lofi/1            33455       9   30101     1%    /mnt
# ls /mnt
./           ../          lost+found/
# umount /mnt
# lofiadm -d /dev/lofi/1
```

例 3　UNIX ファイル上の PC (FAT) ファイルシステムの作成

次に示す一連のコマンドは、FAT ファイルシステムを UNIX ファイル上に作成します。ファイルは lofiadm によって作成されたブロックデバイスに関連付けられています。

```
# mkfile 10M /export/test/testfs
# lofiadm -a /export/test/testfs
/dev/lofi/1
Note use of rlofi, not lofi, in following command.
# mkfs -F pcfs -o nofdisk,size=20480 /dev/rlofi/1
Construct a new FAT file system on /dev/rlofi/1: (y/n)? y
# mount -F pcfs /dev/lofi/1 /mnt
# cd /mnt
# df -k .
Filesystem            kbytes    used   avail capacity  Mounted on
```

例 3　UNIX ファイル上の PC (FAT) ファイルシステムの作成　　*(続き)*

```
/dev/lofi/1            10142       0   10142     0%    /mnt
```

例 4　既存の CD-ROM イメージの圧縮

次の例では、既存の CD-ROM イメージ (`solaris.iso`) を圧縮し、イメージが圧縮されたことを確認してから、圧縮を解除します。

```
# lofiadm -C gzip /export/home/solaris.iso
```

次のように `lofiadm` を使用して、これにブロックデバイスを接続します。

```
# lofiadm -a /export/home/solaris.iso
  /dev/lofi/1
```

マップされたイメージが圧縮されたかどうかを確認します。

```
# lofiadm
Block Device      File                            Options
/dev/lofi/1       /export/home/solaris.iso        Compressed(gzip)
/dev/lofi/2       /export/home/regular.iso        -
```

圧縮されたイメージをマップ解除して、圧縮を解除します。

```
# lofiadm -d /dev/lofi/1
# lofiadm -U /export/home/solaris.iso
```

例 5　暗号化 UFS ファイルシステムをファイル上に作成する

この例は、前述の UFS ファイルシステムの作成例に似ています。

ファイルを作成します。

```
# mkfile 35m /export/home/test
```

ファイルをブロックデバイスに接続して、ファイルイメージの暗号化を指定します。このコマンドの結果、文字デバイスが取得されます。この文字デバイスは、あとで `newfs` により使用されます。

```
# lofiadm -c aes-256-cbc -a /export/home/secrets
Enter passphrase: My-M0th3r;l0v3s_m3+4lw4ys!           (not echoed)
Re-enter passphrase: My-M0th3r;l0v3s_m3+4lw4ys!        (not echoed)
/dev/lofi/1

# newfs /dev/rlofi/1
newfs: construct a new file system /dev/rlofi/1: (y/n)? y
/dev/rlofi/1:   71638 sectors in 119 cylinders of 1 tracks, 602 sectors
       35.0MB in 8 cyl groups (16 c/g, 4.70MB/g, 2240 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 9664, 19296, 28928, 38560, 48192, 57824, 67456,
```

マップされたファイルシステムは、暗号化が有効であることを示します。

例 5　暗号化 UFS ファイルシステムをファイル上に作成する　　*(続き)*

```
# lofiadm
Block Device    File                      Options
/dev/lofi/1     /export/home/secrets      Encrypted
```

ファイルシステムをマウントおよび使用します。

```
# mount /dev/lofi/1 /mnt
# cp moms_secret_*_recipe /mnt
# ls /mnt
./           moms_secret_cookie_recipe    moms_secret_soup_recipe
../          moms_secret_fudge_recipe     moms_secret_stuffing_recipe
lost+found/  moms_secret_meatloaf_recipe  moms_secret_waffle_recipe
# umount /mnt
# lofiadm -d /dev/lofi/1
```

以後、不正な鍵または不正な暗号化アルゴリズムを使用してファイルシステムをマップしようとしても失敗します。

```
# lofiadm -c blowfish-cbc -a /export/home/secrets
Enter passphrase: mommy                                (not echoed)
Re-enter passphrase: mommy                             (not echoed)
lofiadm: could not map file /root/lofi: Invalid argument
# lofiadm
Block Device    File                     Options
#
```

暗号化なしでファイルシステムのマップを試みると成功します。ただし、ファイルシステムをマウントして使用しようとしても失敗します。

```
# lofiadm -a /export/home/secrets
/dev/lofi/1
# lofiadm
Block Device    File                      Options
/dev/lofi/1     /export/home/secrets      -
# mount /dev/lofi/1 /mnt
mount: /dev/lofi/1 is not this fstype
#
```

環境変数

lofiadm の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。LC_CTYPE、LC_MESSAGES、および NLSPATH。

終了ステータス

次の終了値が返されます。

0
　正常終了。

>0
　エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

fsck(1M), mount(1M), mount_ufs(1M), newfs(1M), zonecfg(1M), attributes(5), resource_controls(5), lofi(7D), lofs(7FS)

注意事項

マウントされたファイルシステムを持つディスクデバイスに直接アクセスしないのと同様に、lofi ファイルドライバを使用する場合を除き、ブロックデバイスに関連付けられているファイルにアクセスしないようにしてください。また、そのようなアクセスを防止するために、適切なアクセス権をファイルに設定することもお勧めします。

lofiadm の機能およびこの機能を使用できるユーザーは、/dev/lofictl のアクセス権によって制御されます。読み取りアクセス権は、すべての関連付けの表示などのクエリー操作を可能にします。書き込みアクセス権は、関連付けの追加などの状態変更操作の実行に必要です。出荷時の /dev/lofictl は、root によって所有され、グループ sys に所属し、モード 0644 に設定されているため、すべてのユーザーがクエリー操作を実行できますが、変更操作はルートのみに可能です。管理者は、ユーザーに書き込みアクセスを与えて、関連付けの追加または削除を許可することができますが、これはセキュリティーホールになる可能性が非常に高いため、信頼できるグループにのみ与えるようにしてください。

ファイルシステムイメージをマウントする場合は、適切なマウントオプションを使用するように注意してください。特に、出所が不明な UFS イメージには、nosuid マウントオプションが適切な場合があります。また、UFS に logging や forcedirectio を使用する場合のように、いくつかのオプションは役に立たなかったり適切でなかったりすることがあります。互換性を保つために、raw デバイスもブロックデバイスと一緒にエクスポートされます。たとえば、newfs(1M) にはこれが必要です。

lofiadm (引数なし) の出力は、将来のリリースで変更される可能性があります。

名前

logadm – manage endlessly growing log files

形式

```
logadm
logadm [-options] logname...
```

機能説明

logadm is a general log rotation tool that is suitable for running from cron(1M).

Without arguments, logadm reads the /etc/logadm.conf file, and for every entry found in that file checks the corresponding log file to see if it should be rotated. Typically this check is done each morning by an entry in the root's crontab.

If the *logname* argument is specified, logadm renames the corresponding log file by adding a suffix so that the most recent log file ends with .0 (that is, *logfile*.0), the next most recent ends with .1 (that is, *logfile*.1), and so forth. By default, ten versions of old log files are kept (that is, *logfile*.0 through *logfile*.9). At the point when what would be the eleventh file is logged, logadm automatically deletes the oldest version to keep the count of files at ten.

logadm takes a number of *options*. You can specify these options on the command line or in the /etc/logadm.conf file. The logadm command searches /etc/logadm.conf for lines of the form *logname options*

logname
: Identifies an entry in /etc/logadm.conf. This can be a name or the pathname of the log file. If you specify a log file, rather than a name, for this field, it must be a fully qualified pathname.

options
: Identifies command line options exactly as they would be entered on the command line. This allows commonly used log rotation policies to be stored in the /etc/logadm.conf file. See EXAMPLES.

  If *options* are specified both in /etc/logadm.conf and on the command line, those in the /etc/logadm.conf file are applied first. Therefore, the command line options override those in /etc/logadm.conf.

  Log file names specified in /etc/logadm.conf may contain filename substitution characters such as * and ?, that are supported by csh(1).

Two options control when a log file is rotated. They are: -s size -p period.

When using more than one of these options at a time, there is an implied *and* between them. This means that all conditions must be met before the log is rotated.

If neither of these two options are specified, the default conditions for rotating a log file are: -s 1b -p 1w, which means the log file is only rotated if the size is non-zero and if at least 1 week has passed since the last time it was rotated.

By specifying `-p never` as a rotation condition, any other rotation conditions are ignored and `logadm` moves on to the expiration of old log files. By specifying `-p now` as a rotation condition, a log rotation is forced.

Unless specified by the `-o`, `-g`, or `-m` options, `logadm` replaces the log file (after renaming it) by creating an empty file whose owner, group ID, and permissions match the original file.

Three options control when old log files are expired: `-A` age `-C` count `-S` size. These options expire the oldest log files until a particular condition or conditions are met. For example, the combination `-C 5` and the `-S 10m` options expires old log files until there are no more than 5 of the *and* their combined disk usage is no more than 10 megabytes. If none of these options are specified, the default expiration is `-C 10` which keeps ten old log files. If no files are to be expired, use `-C 0` to prevent expiration by default.

`logadm` stores timestamps in the file `/var/logadm/timestamps`. For users of previous versions of `logadm`, the utility automatically moves timestamps from `/etc/logadm.conf`, their previous repository, to `/var/logadm/timestamps`.

オプション

The following options are supported:

`-a` *post_command*
: Execute the *post_command* after renaming the log file. *post_command* is passed to `sh -c`.

  Specify *post_command* as a valid shell command. Use quotes to protect spaces or shell metacharacters in *post_command*.

  This option can be used to restart a daemon that is writing to the file. When rotating multiple logs with one `logadm` command, *post_command* is executed only once after all the logs are rotated, not once per rotated log.

`-A` *age*
: Delete any versions that have not been modified for the amount of time specified by *age*.

  Specify *age* as a number followed by an `h` (hours), `d` (days), `w`(weeks), `m` (months), or `y` (years).

`-b` *pre_command*
: Execute *pre_command* before renaming the log file. *pre_command* is passed to `sh -c`.

  Specify *pre_command* as a valid shell command. Use quotes to protect spaces or shell metacharacters in the *pre_command*.

  This option can be used to stop a daemon that is writing to the file. When rotating multiple logs with one `logadm` command, *pre_command* is executed only once before all the logs are rotated, not once per rotated log.

`-c`
: Rotate the log file by copying it and truncating the original logfile to zero length, rather than renaming the file.

-C *count*
Delete the oldest versions until there are not more than *count* files left.

If no expire options (-A, -C, or -S) are specified, -C 10 is the default. To prevent the default expire rule from being added automatically, specify -C 0 .

-e *mail_addr*
Send error messages by email to *mail_addr*.

As logadm is typically run from cron(1M), error messages are captured by cron and mailed to the owner of the crontab.

This option is useful if you want the mail regarding error messages to go to another address instead. If no errors are encountered, no mail message is generated.

-E *cmd*
Execute *cmd* to expire the file, rather than deleting the old log file to expire it.

cmd is passed it to sh -c. The file is considered expired after *cmd* completes. If the old log file is not removed or renamed by the *cmd*, logadm considers it for expiration the next time that it runs on the specified log file. If present, the keyword $file is expanded in the specified *cmd*to the name of the file being expired.

This option is useful for tasks such as mailing old log files to administrators, or copying old log files to long term storage.

-f *conf_file*
Use *conf_file* instead of /etc/logadm.conf.

This option allows non-root users to keep their own logadm configuration files.

-F *timestamp_file*
Use *timestamp_file* instead of /var/logadm/timestamps to store logadm timestamps.

-g *group*
Create a new empty file with the ID specified by *group*, instead of preserving the group ID of the log file.

Specify *group* by name or by numeric group ID, as accepted by chgrp(1).

This option requires the ability to change file group ownership using the chgrp(1) command.

-h
Print a help message that describes logadm's options.

-l
Use local time rather than the Coordinated Universal Time (UTC) when naming rotated log files (see the discussion of percent sequences in the templates supplied with the -t option).

`-m` *mode*
Create a new empty file with the mode specified by *mode*, instead of preserving the mode of the log file.

Specify *mode* in any form that is accepted by the `chmod(1)` command.

`-M` *cmd*
Use *cmd* to rename the log file. If the keyword `$file` is specified, it is expanded to the name of the log file. Similarly, the keyword `$nfile` is expanded to the new name of the log file. The `$nfile` keyword is only available with commands provided with the `-M` option. After the command completes, the log file is replaced by the rotate file. The default *cmd* is "`/bin/mv $file$nfile`".

`-n`
Print the actions that the `logadm` command will perform without actually performing them.

This option is useful for checking arguments before making any changes to the system.

It is important to remember, however, that since log rotating actions are only printed with this option, `logadm` might not find files that need expiring, but if run without the `-n` `logadm` might create a file that needs expiring by performing the log rotating actions. Therefore, if you see no files being expired with the `-n` option, files still might be expired without it.

`-N`
Prevent an error message if the specified logfile does not exist. Normally, `logadm` produces an error message if the log file is not found. With `-N`, if the log file doesn't exist `logadm` moves on to the expire rules (if any) and then to the next log file (if any), without creating the empty replacement log file.

`-o` *owner*
Create the new empty file with *owner*, instead of preserving the owner of the log file.

Specify *owner* in any form that is accepted by the `chown(1)` command.

`-p` *period*
Rotate a log file after the specified time period (*period*).

Specify *period* as a number followed by `d` for days, `h` for hours, `w` for weeks, `m` for months (30 days) or `y` for years. There are also two special values for period: `now` and `never`. "`-p now`" forces log rotation. "`-p never`" forces no log rotation.

`-P` *timestamp*
Used by `logadm` to record the last time the log was rotated in `/var/logadm/timestamps`.

This option uses *timestamp* to determine if the log rotation period has passed. The format of *timestamp* matches the format generated by `ctime(3C)`, with quotes around it to protect embedded spaces. *timestamp* is always recorded in the Coordinated Universal Time (UTC) timezone.

`-r`
Remove any entries corresponding to the specified *logname* from the `/etc/logadm.conf`.

`-R` *cmd*
Run the *cmd* when an old log file is created by a log rotation. If the keyword `$file` is embedded in the specified command, it is expanded to the name of the old log file just created by log rotation.

This option is useful for processing log file contents after rotating the log. *cmd* is executed by passing it to `sh -c`. When rotating multiple logs with one logadm command, the command supplied with `-R` is executed once every time a log is rotated. This is useful for post-processing a log file (that is, sorting it, removing uninteresting lines, etc.). The `-a` option is a better choice for restarting daemons after log rotation.

`-s` *size*
Rotate the log file only if its size is greater than or equal to *size*.

Specify *size* as a number followed by the letter `b` for bytes, `k` for kilobytes, `m` for megabytes, or `g` for gigabytes.

`-S` *size*
Delete the oldest versions until the total disk space used by the old log files is less than the specified size.

Specify *size* as a number followed by the letter `b` for bytes, `k` for kilobytes, `m` for megabytes, or `g` for gigabytes.

`-t` *template*
Specify the template to use when renaming log files.

template can be a simple name, such as `/var/adm/oldfile`, or it can contain special keywords which are expanded by `logadm` and are in the form `$`*word*. Allowed sequences are:

$*basename*
The log file name, without the directory name

$*dirname*
The directory of the file to be rotated

$*domain*
Expands to the output of `domainname`

$*file*
The full path name of the file to be rotated

$*isa*
Expands to the output of `uname -p`

$*machine*
Expands to the output of `uname -m`

$*n*
The version number, `0` is most recent, `1` is next most recent, and so forth

$*N*
The same as $*n*, but starts at `1` instead of zero

$*nodename*
Expands to the output of `uname -n`

$*platform*
Expands to the output of `uname -i`

$*release*
Expands to the output of `uname -r`

$*secs*
The number of seconds since `00:00:00 UTC, January 1,1970`

$*zonename*
Expands to the output of `zonename(1)`.

To actually have the dollar sign character in the file name, use `$$`. Any percent sequences allowed by `strftime(3C)` are also allowed, for example, `%d` expands to the day of the month. To actually have a percent sign character in the file name, use `%%`. Both dollar-sign keywords and percent sequences can appear anywhere in the template. If the template results in a pathname with non-existent directories, they are created as necessary when rotating the log file.

If no `-t` option is specified, the default template is `$file.$n`. Actual *rotation* of log files, where each version is shifted up until it expires is done using the `$n` keyword. If the template does not contain the `$n` keyword, the log file is simply renamed to the new name and then the expire rules, if any, are applied.

`-T` *pattern*
Normally `logadm` looks for a list of old log files by turning the template (specified with the `-t` option) into a pattern and finding existing files whose names match that pattern. The `-T` option causes the given pattern to be used instead.

This option is useful if another program fiddles with the old log file names, like a `cron` job to compress them over time. The pattern is in the form of a pathname with special characters such as `*` and `?` as supported by `csh(1)` filename substitution.

`-v`
Print information about the actions being executed in verbose mode.

`-V`
Validate the configuration file.

This option validates that an entry for the specified *logname* exists in the `/etc/logadm.conf` file and is syntactically correct. If *logname* is not specified, all entries in

the configuration file are validated. If a `logname` argument is specified, the command validates the syntax of that entry. If the entry is found, it is printed and the exit value of the command is true. Otherwise the exit value is false.

`-w` *entryname*
: Write an entry into the config file (that is, `/etc/logadm.conf`) that corresponds to the current command line arguments. If an entry already existed for the specified *entryname*, it is removed first. This is the preferred method for updating `/etc/logadm.conf`, because it prevents syntax errors. The *entryname* is an argument to an invocation of `logadm`. *entryname* might be chosen as something easy to remember or it can be the pathname of the log file. If a pathname, rather than a name is used, it must be a fully qualified pathname.

If no log file name is provided on a `logadm` command line, the entry name is assumed to be the same as the log file name. For example, the following two lines achieve the same thing, keeping two copies of rotated log files:

```
% logadm -C2 -w mylog /my/really/long/log/file/name
% logadm -C2 -w /my/really/long/log/file/name
```

`-z` *count*
: Compress old log files after all other commands have been executed. *count* of the most recent log files are left uncompressed, therefore making the *count* most recent files easier to peruse. Use *count* of zero to compress all old logs.

The compression is done with `gzip`(1) and the resulting log file has the suffix of `.gz`.

オペランド

The following operands are supported:

logname
: Identifies the name of the entry in `/etc/logadm.conf`. If the log file name is specified in the *logname* field, it is assumed that *logname* is the same as the actual log file name.

使用例

例 1 Rotating a File and Keeping Previous Versions

The following example rotates the `/var/adm/exacct/proc` file, keeping ten previous versions in `/var/adm/exacct/proc.0` through `/var/adm/exacct/proc.9`.

Tell `logadm` to copy the file and truncate it.

```
% logadm -c /var/adm/exacct/proc
```

例 2 Rotating syslog

The following example rotates `syslog` and keeps eight log files. Old log files are put in the directory `/var/oldlogs` instead of `/var/log`:

```
% logadm -C8 -t'/var/oldlogs/syslog.$n' /var/log/syslog
```

例 3 Rotating /var/adm/sulog and Expiring Based on Age

The following entry in the /etc/logadm.conf file rotates the /var/adm/sulog file and expires any copies older than 30 days.

```
/var/adm/sulog -A 30d
```

例 4 Rotating Files and Expiring Based on Disk Usage

The following entry in the /etc/logadm.conf file rotates the /var/adm/sulog file and expires old log files when more than 100 megabytes are used by the sum of all the rotated log files.

```
/var/adm/sulog -S 100m
```

例 5 Creating an Entry that Stores the Logfile Name

This example creates an entry storing the log file name and the fact that we want to keep 20 copies in /etc/logadm.conf, but the -p never means the entry is ignored by the normal logadm run from root's crontab every morning.

```
% logadm -w locallog /usr/local/logfile -C20 -p never
```

Use the following entry on the command line to override the -p never option:

```
 % logadm -p now locallog
```

例 6 Rotating the apache Error and Access Logs

The following example rotates the apache error and access logs monthly to filenames based on current year and month. It keeps the 24 most recent copies and tells apache to restart after renaming the logs.

This command is run once, and since the -w option is specified, an entry is made in /etc/logadm.conf so the apache logs are rotated from now on.

```
 % logadm -w apache -p 1m -C 24\
     -t '/var/apache/old-logs/$basename.%Y-%m'\
     -a '/usr/apache/bin/apachectl graceful'\
     '/var/apache/logs/*{access,error}_log'
```

This example also illustrates that the entry name supplied with the -w option doesn't have to match the log file name. In this example, the entry name is apache and once the line has been run, the entry in /etc/logadm.conf can be forced to run by executing the following command:

```
% logadm -p now apache
```

Because the expression matching the apache log file names was enclosed in quotes, the expression is stored in /etc/logadm.conf, rather than the list of files that it expands to. This means that each time logadm runs from cron it expands that expression and checks all the log files in the resulting list to see if they need rotating.

例 6 Rotating the apache Error and Access Logs (続き)

The following command is an example without the quotes around the log name expression. The shell expands the last argument into a list of log files that exist at the time the command is entered, and writes an entry to /etc/logadm.conf that rotates the files.

```
logadm -w apache /var/apache/logs/*_log
```

ファイル

/etc/logadm.conf
: configuration file for logadm command

/var/logadm/timestamps
: repository for logging timestamps

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

chgrp(1), chmod(1), chown(1), csh(1), gzip(1), cron(1M), ctime(3C), strftime(3C), logadm.conf(4), attributes(5)

注意事項

When logadm applies expire conditions (supplied by the -A, -C, and -S options), it deletes files, the oldest first, until the conditions are satisfied. If the template used for naming the old logs contained $n or $N, logadm picks the highest value of $n or $N found in the old log file names first. If the template used is something else, logadm uses the modification time to determine which files to expire first. This may not be the expected behavior if an old log file has been modified since it was rotated.

Depending on log file sizes and number of log files, log file rotations can be very time-consuming.

By default, logadm works in UTC. Therefore, all entries written to the /etc/logadm.conf file (see logadm.conf(4)) will have a UTC timestamp. Users can use the -l option to set logadm to local time.

The -f and -F options can specify the same file, in which case logadm reverts to the same behavior as in prior releases. That is, timestamps are written to the configuration file.

名前

logins – list user and system login information

形式

```
/usr/bin/logins [-admopstux] [-g group...]
     [-l login_name...]
```

機能説明

This command displays information on user and system logins known to the system. Contents of the output is controlled by the command options and can include the following: user or system login, user id number, `passwd` account field value (user name or other information), primary group name, primary group id, multiple group names, multiple group ids, home directory, login shell, and four password aging parameters. The default information is the following: login id, user id, primary group name, primary group id and the account field value. Output is sorted by user id, system logins, followed by user logins.

オプション

Options may be used together. If so, any login that matches any criteria are displayed.

The following options are supported:

`-a`
: Add two password expiration fields to the display. The fields show how many days a password can remain unused before it automatically becomes inactive, and the date that the password expires.

`-d`
: Selects logins with duplicate uids.

`-g` *group*
: Selects all users belonging to `group`, sorted by login. Multiple groups can be specified as a comma-separated list. When the `-l` and `-g` options are combined, a user is only listed once, even if the user belongs to more than one of the selected groups.

`-l` *login_name...*
: Selects the requested login. Multiple logins can be specified as a comma-separated list. Depending on the nameservice lookup types set in `/etc/nsswitch.conf`, the information can come from the `/etc/passwd` and `/etc/shadow` files and other nameservices. When the `-l` and `-g` options are combined, a user is only listed once, even if the user belongs to more than one of the selected groups.

`-m`
: Displays multiple group membership information.

`-o`
: Formats output into one line of colon-separated fields.

`-p`
: Selects logins with no passwords.

`-s`
: Selects all system logins.

`-t`
: Sorts output by login instead of by uid.

-u
Selects all user logins.

-x
Prints an extended set of information about each selected user. The extended information includes home directory, login shell and password aging information, each displayed on a separate line. The password information currently consists of password status:

NP
Account has no password

LK
Account is locked for UNIX authentication

NL
Account is a no login account

UP
This account has not yet been activated by the administrator and cannot be used.

PS
Account probably has a valid password

UN
Account password status is unknown. That is, it is not a recognizable hashed password or any of the above entries. See crypt(3C) for valid password hashes.

If the login is passworded, status is followed by the date the password was last changed, the number of days required between changes, and the number of days allowed before a change is required. The password aging information shows the time interval that the user receives a password expiration warning message (when logging on) before the password expires.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

passwd(1), crypt(3C), attributes(5)

名前

lpget – get printing configuration

形式

`lpget [-k key] [destination... | list]`

機能説明

The `lpget` utility reads printing configuration information from the configuration databases in `$HOME/.printers` and `printers.org_dir` printer. This information, called a *configuration report*, is displayed to the standard output.

`lpget` displays a configuration report for all keys for the specified destination or destinations by default. Use the `-k` option to display a configuration report for specific keys. Use the `list` operand to display a configuration report for all configured destinations.

オプション

The following option is supported:

`-k` *key* Displays a configuration report for *key*.

オペランド

The following operands are supported:

destination Displays a configuration report for *destination*. Destination can be either a printer of a class of printers. Specify *destination* using atomic or POSIX-style (*server*:*destination*) names. See `standards(5)` for information concerning POSIX.

`list` Displays a configuration report for all configured destinations.

使用例

例 1 Displaying a Configuration Report for the bsdaddr Key

The following example displays a configuration report for the `bsdaddr` key for printer `catalpa`.

```
example% lpget -k bsdaddr catalpa
```

例 2 A Configuration Report for all Keys for all Configured Destinations

The following example displays a configuration report for all keys for all configured destinations.

```
example% lpget list
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

non-zero An error occurred.

ファイル

`$HOME/.printers` User-configurable printer database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/lp/print-client-commands
Interface Stability	Committed

関連項目

lpset(1M), attributes(5), standards(5), ldap(5)

『Oracle Solaris Administration: Naming and Directory Services』

注意事項

If the LDAP database is used as the name service. If the ldapclient(1M) server is a replica LDAP server, LDAP printer database updates may not appear immediately, as the replica server may not have been updated by the master server and can be out of sync. For example, a printer that you deleted by using lpset(1M) may still appear in the printer list you display with lpget until the replica is updated from the master. Replica servers vary as to how often they are updated from the master. Refer to the 『Oracle Solaris Administration: Naming and Directory Services』 for more information on LDAP replication.

名前

lpset – set printing configuration in supported databases

形式

```
lpset [-n system | fnsldap] [-x]
     [ [-D binddn] [-w passwd] [-h ldaphost]]
     [-a key=value] [-d key] destination
```

機能説明

The `lpset` utility sets printing configuration information in the system configuration databases. See `nsswitch.conf(4)`.

Only a superuser or a member of Group 14 may execute `lpset`.

オプション

The following options are supported:

`-n ldap`	Create or update the configuration information for the *destination* entry in LDAP printer contexts. `ldap` specifies that the information is written to an LDAP server. See `NOTES`.
`-x`	Remove all configuration for the *destination* entry from the database specified by the `-n` option.
`-a` *key=value*	Configure the specified *key=value* pair for the *destination*.
`-d` *key*	Delete the configuration option specified by *key* for the *destination* entry.
`-D binddn`	Use the distinguished name (DN) `binddn` to bind to the LDAP directory server.
`-w passwd`	Use `passwd` as the password for authentication to the LDAP directory server.
`-h ldaphost`	Specify an alternate host on which the LDAP server is running. This option is only used when `ldap` is specified as the naming service. If this option is not specified, the default is the current host system.

オペランド

The following operand is supported:

destination	Specifies the entry in LDAP, in which to create or modify information. *destination* names a printer of class of printers. Specify *destination* using atomic names. POSIX-style destination names are not acceptable. See `standards(5)` for information regarding POSIX.

使用例

例 1　Setting a key=value Pair

The following example sets the `user-equivalence =true` *key=value* pair for destination `tabloid` in the NIS context:

```
example% lpset -n nis -a user-equivalence=true tabloid
```

例 2　Setting a key=value Pair in LDAP

```
example% lpset -n ldap -h ldapl.xyz.com -D "cn=Directory Manager" \
   -w passwd -a key1=value1 printer1
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

non-zero An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/lp/print-client-commands
Interface Stability	Committed

関連項目

`ldapclient(1M)`, `lpget(1M)`, `nsswitch.conf(4)`, `attributes(5)`, `ldap(5)`, `standards(5)`

『Oracle Solaris Administration: Naming and Directory Services』

注意事項

If the `ldap` database is used, the printer administrator should be mindful of the following when updating printer information.

1. Because the domain information for the printer being updated is extracted from the `ldapclient(1M)` configuration, the LDAP server being updated must host the same domain that is used by the current `ldapclient(1M)` server.
2. If the LDAP server being updated is a replica LDAP server, the updates will be referred to the master LDAP server and completed there. The updates might be out of sync and not appear immediately, as the replica server may not have been updated by the master server. For example, a printer that you deleted by using `lpset` may still appear in the printer list you display with `lpget` until the replica is updated from the master. Replica servers vary as to how often they are updated from the master.
3. Although users can use the LDAP command line utilities `ldapadd(1)` and `ldapmodify(1)` to update printer entries in the directory, the preferred method is to use `lpset`. Otherwise, if the `ldapadd` and `ldapmodify` utilities are used, the administrator must ensure that the `printer-name` attribute value is unique within the `ou=printers` container on the LDAP server. If the value is not unique, the result of modifications done using `lpset` might be unpredictable.

名前

luxadm – Sun Fire 880 記憶装置サブシステムと FC_AL デバイスの管理

形式

```
luxadm [options]... subcommand [options]... enclosure
      [,dev] | pathname...
```

機能説明

`luxadm` プログラムは、SENA、Sun Fire 880 内蔵記憶装置サブシステム、および各 FC_AL (Fiber Channel Arbitrated Loop) デバイスの管理コマンドです。`luxadm` は、コマンド行に指定される引数やオプションに応じて、さまざまな制御処理や照会処理を実行します。

コマンド行にはサブコマンドの指定が必要です。コマンド行には、サブコマンドに応じてオプションやパラメータを指定することができます。オプションには、通常 1 つ以上の格納装置名またはパス名を指定します。指定する文字数は、サブコマンドを一意に識別するのに必要なだけでかまいません。

サブコマンドが作用するデバイスをパス名で指定します。Sun StorEdge A5000 サブシステムでは、パス名の代わりに、デバイスまたはデバイスに対するポートのワールドワイド名 (WWN) を入力することによってディスクデバイスまたは格納装置サービスコントローラを指定することができます。また、Sun StorEdge A5000 の格納装置名および格納装置内の特定のデバイスを示す識別子 (オプション) を入力することによってデバイスを指定することもできます。各 FC_AL デバイスを指定するには、デバイスの WWN またはポートを入力します。

パス名

デバイスまたはコントローラを、完全な物理パス名または完全な論理パス名で指定します。

Sun StorEdge A5000 に対するデバイスの一般的な物理パス名を以下に示します。

```
/devices/sbus@1f,0/SUNW,socal@1,0/sf@0,0/ssd@w2200002037000f96,
        0:a,raw
```

システム上のすべての Sun StorEdge A5000 IB (Interface Board)に対しては、物理パスへの論理リンクが `/dev/es` ディレクトリに格納されます。論理リンクの例として、`/dev/es/ses0` などがあります。

FC_AL デバイスや Sun StorEdge A5000 サブシステム IB を選択するために、パス名の代わりに WWN を使用することもできます。WWN は、デバイスを使用するためのポートまたはデバイスそのものを表す 16 桁の 16 進数の値です。一般的な WWN の値を次に示します。

```
2200002037000f96
```

WWN の形式については、「注意事項」を参照してください。

Sun Fire 880 内蔵記憶装置サブシステムのディスクの一般的な物理パス名を次に示します。

```
/devices/pci@8,600000/SUNW,qlc@2/fp@0,0/ssd@w2100002037a6303c,0:a
```

次に、一般的な論理パス名を示します。

```
/dev/rdsk/c2t8d0s2
```

各 FC_AL デバイスの一般的なパス名を示します。

```
/devices/sbus@3.0/SUNW,socal@d,10000/sf@0,0/ssd@w2200002037049fc3,0:a,raw
```

次に、一般的な論理パス名を示します。

```
/dev/rdsk/c1t0d0s2
```

格納装置

Sun StorEdge A5000 では、デバイスは格納装置名とスロット名で特定することができます。

box_name[,f*slot_number*]
box_name[,r*slot_number*]

box_name は、Sun StorEdge A5000 の格納装置名で、`enclosure_name` サブコマンドで指定します。オプションの *slot_number* パラメータを指定しないと、*box_name* には、Sun StorEdge A5000 サブシステム IB が指定されます。

`f` または `r` は、Sun StorEdge A5000 の格納装置の前面のスロット(f) か、背面のスロット (r) かを指定します。

slot_number は、Sun StorEdge A5000 の格納装置のデバイスのスロット番号を指定します。指定することができる番号の範囲は `0`〜 `6` または `0` 〜 `10` です。

Sun Fire 880 内蔵記憶装置サブシステムも、デバイスは格納装置名とスロット名で特定することができます。ただし、1 セットのディスクしかありません。

box_name[,s*slot_number*]

box_name は、Sun Fire 880 内蔵記憶装置サブシステムの格納装置名で、`enclosure_name` サブコマンドで指定します。オプションの *slot_number* パラメータを指定しないで使用した場合、*box_name* には、Sun Fire 880 内蔵記憶装置サブシステム格納装置のサービスデバイスが指定されます。Sun Fire 880 内蔵記憶装置サブシステムのスロット番号、`0` 〜 `11` のいずれかを指定するには *s* を使用します。

ディスクおよびサブシステムの論理名については、`disks(1M)` および `devlinks(1M)` を参照してください。

オプション

次のオプションはすべてのサブコマンドに対して指定することができます。

`-e` エキスパート (Expert) モードです。このオプションは初心者の方にはお勧めしません。

`-v` 冗長モード。

特定のサブコマンドだけに指定するオプションについては、「使用方法」の各サブコマンドの説明を参照してください。

オペランド　次のオペランドがサポートされています。

enclosure
Sun StorEdge A5000 の *box_name* を指定します。

fibre_channel_HBA_port
ホストのコントローラポートへのパスを指定します。一般的なパスは、下記のとおりです。

```
/devices/pci@8,600000/pci@1/SUNW,qlc@4/fp@0,0:devctl
```

pathname
SENA IB、Sun Fire 880 内蔵記憶装置サブシステム、またはディスクデバイスの論理パスまたは物理パスを指定します。*pathname* は、SENA IB、SENA ディスク、または各 FC_AL デバイスの WWN でも指定することができます。

使用法

サブコマンド

`display` *enclosure*[,*dev*] . . . | *pathname* . . .
`display -p` *pathname* . . .
`display -r` *enclosure*[,*dev*] . . . | *pathname* . . .
`display - v` *enclosure*[,*dev*] . . . | *pathname* . . .
格納装置またはデバイス固有のデータを表示します。

サブシステムのデータは、格納装置の環境検知情報、およびディスクのステータスを含む、すべてのサブシステムのデバイスのステータスで構成されます。

ディスクデータは、照会、容量、および設定情報で構成されます。

`-p`　*pathname* で指定されたデバイスまたはサブシステムの性能情報を表示します。このオプションは、性能情報を保存するサブシステムに対してのみ指定することができます。

`-r`　pathname で指定された FC_AL デバイスのエラー情報を表示します。また、パス名が Sun StorEdge A5000 の場合は、そのループ上のすべてのデバイスのエラー情報を表示します。`-r` オプションは Sun StorEdge A5000 サブシステムまたは各 FC_AL デバイスに対してのみ指定することができます。

`-v`　モード検知データを含む、詳細表示モードで表示します。

`download` [`-s`] [`-f` *filename_path*] *enclosure*. . .
prom イメージを、filename_path で指定された SENA サブシステムのインタフェースボード装置か、enclosure または pathname で指定された SPARCstorageArray コントローラにダウンロードします。SPARCstorage Array は、ダウンロードコードを使用するためにリセットする必要があります。

SENA のダウンロードが終了すると、SENA はリセットされ、ダウンロードコードが実行されます。ファイル名が指定されていない場合は、デフォルトの prom イメージが使用されます。SPARCstorage Array コントローラ用のデフォルト

の prom イメージは、/usr/lib/firmware/ssa/ssafirmware ディレクトリに格納されます。SENA 用のデフォルトの prom イメージは `usr/lib/locale/C/LC_MESSAGES` ディレクトリに `ibfirmware` というファイル名で格納されます。

Sun Fire 880 内蔵記憶装置サブシステムのダウンロードが終了すると、サブシステムはリセットされ、ダウンロードコードが実行されます。Sun Fire 880 内蔵記憶装置サブシステムのデフォルトのファームウェアイメージは、次のディレクトリに格納されます。

`/usr/platform/SUNW,Sun-Fire-880/lib/images/int_fcbpl_fw`。

`-s` 保存オプションです。`-s` オプションはダウンロードしたファームウェアを FEPROM に保存します。`-s` を指定しない場合、 ダウンロードしたファームウェアは保存されず、電源の再投入後に消えてしまいます。

`-s` オプションは SPARCstorage Array コントローラには指定することができません。

`-s` オプションを使用すると、ダウンロードサブコマンドがサブシステム上の FEPROMを変更するため、-s オプションの使用には注意が必要です。

`enclosure_name` *new_name enclosure* | *pathname*

enclosure または pathname で指定された単数または複数の格納装置名を変更します。新しい名前は 16 文字以下で *new_name* に指定します。英数字のみが使用できます。このサブコマンドは Sun StorEdge A5000 に対してだけ指定することができます。

`failover primary` | `secondary` *pathname*

指定した論理ボリュームに、どの Sun StorEdge T3 array パートナーグループのコントローラがアクセスするかを選択します。`primary` を指定した場合、プライマリコントローラから論理ボリュームにアクセスします。`secondary` を指定した場合、*pathname* で指定したセカンダリコントローラから論理ボリュームにアクセスします。

`fcal_s_download` [`-f` *fcode-file*]

fcode-file で指定されたファイルに含まれる fcode をすべての FC/S SBus カードにダウンロードします。このコマンドは対話型で、fcode をダウンロードする前にユーザーに対して確認を促します。

`fcal_s_download` は、シングルユーザーモードでだけ使用してください。入出力操作が行われているホストアダプタに対して、そのアダプタを更新する目的で `fcal_s_download` を指定すると、アダプタのリセットの原因となります。新しく更新した fcode は、システムをリブートしたときに実行され、表示できるようになります。

`-f` *fcode-file* `-f` fcode-file オプションを省略すると、各 FC100/S SBus カードの中の現在のバージョンの fcode が表示されます。

`fcode_download -p`
`fcode_download -d` *dir-name*

インストール済みの FC/S、FC100/S、FC100/P、または FC100/2P ホストバスアダプタ カードを検出し、*dir-name* の fcode を適切なカードにダウンロードします。このコマンドは対話型で、各ファイルタイプに対して適切なカードを決定します。各デバイスに fcode をダウンロードする前にユーザーに対して確認を促します。

`fcode_download` は、シングルユーザーモードで fcode だけを読み込むときに使用します。入出力操作があるホストアダプタに対して、そのアダプタを更新する目的で `fcode_download` を指定すると、アダプタのリセットの原因となります。新しく更新した fcode は、システムをリブートしたときに実行され、表示できるようになります。

`-d` *dir-name*
: *dir-name* ディレクトリに格納されている fcode ファイルを適切なアダプタカードにダウンロードします。d オプションを省略すると、デフォルトの usr/lib/firmware/fc_s ディレクトリが使用されます。

`-p`
: 各カードに読み込まれている現在のバージョンの fcode を表示します。ダウンロードは行われません。

`inquiry` *enclosure*[,*dev*] . . . | *pathname* . . .

enclosure または pathname で指定されたデバイスに対する照会情報を表示します。

`insert_device` [*enclosure*,*dev* . . .]

このコマンドを使用して、電源を入れたまま 1 つまたは複数のデバイスを追加することができます。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。このサブコマンドは Sun StorEdge A5000、RSM および各 FC_AL デバイスに対してのみ指定することができます。Sun StorEdge A5000に対して複数の格納装置が指定された場合は、複数のバスに対して同時に追加が行われます。このサブコマンドの引数を省略すると、すべての格納装置または各 FC_AL デバイスが追加されます。RSM に対して指定することができるコントローラは 1 つだけです。Sun StorEdge A5000 に対しては、このサブコマンドは、電源を入れたまま 1 つまたは複数のデバイスを追加するすべての過程をユーザーとの対話によって行います。複数のディスクが指定された場合は、それらのディスクが正しいかどうかの確認が行われ、ユーザーは継続するか中止するかを選択することができます。その後で、ディスクまたは格納装置の追加を実行するかどうかの確認が行われ、それらのデバイスの論理パス名が作成および表示されます。

`led` *enclosure*,*dev* . . . | *pathname*. . .

enclosure または pathname で指定されたディスクに対応している LED の現在の状態を表示します。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`led_blink` *enclosure,dev* . . . | *pathname* . . .
enclosure または pathname で指定されたディスクに対応している LED の点滅を開始するようサブシステムに指示します。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`led_off` *enclosure,dev* . . . | *pathname* . . .
enclosure または pathname で指定されたディスクに対応している LED を消灯させるようサブシステムに指示します。Sun StorEdge A5000 サブシステムでは、LED の消灯や点滅の停止ができる状態とできない状態があります。『Sun StorEdge A5000 設置・サービスマニュアル』(805-4111) を参照してください。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`led_on` *pathname* . . .
pathname で指定されたディスクに対応している LED を点灯させるようサブシステムに指示します。このサブコマンドはこの機能を持つサブシステムに対してのみ指定することができます。

`power_off` [`-F`] *enclosure*[,*dev*] . . . | *pathname* . . .
Sun StorEdge A5000 に対してこのオプションでは、Sun StorEdge A5000 サブシステムが省電力モードに切り替わります。Sun StorEdge A5000 ドライブは、省電力モードでは使用することができません。SPARCstorage Array 中の格納装置サービスカードに対してこのオプションでは、RSMトレーの電源が切れます。Sun StorEdge A5000 中のドライブに対してこのオプションでは、そのドライブがドライブ off/unmated モードに設定されます。ドライブ off/unmated 状態では、ドライブが停止し、バイパスモードになります。このコマンドは、Sun Fire 880 内蔵記憶装置サブシステムには使用できません。

`-F` 強制オプションは Sun StorEdge A5000 のみに適用されます。このオプションを指定すると、`luxadm` は、1 つまたは複数のデバイスを、それらがホストによって使用されていても電源切断しようとします。

警告 : 現在使用されているデータを含むデバイスの電源を切断すると、予想不可能な結果を引き起こします。デバイスの電源を切断する際は、まず、通常の方法を (`-F` を指定せずに) 試してください。このオプションは、通常の確認を無効にすることによる結果を理解した上で使用してください。

`power_on` *enclosure*[, *dev*] . .
Sun StorEdge A5000 に対してこのオプションでは、Sun StorEdge A5000 サブシステムが省電力モードから抜けますSun StorEdge A5000 ドライブは、省電力モードでは使用することができません。SPARCstorage Array RSM トレーの電源をプログラムによって投入する方法はありません。Sun StorEdge A5000 の中のドライブに対してこのオプションでは、そのドライブが通常の起動状態に設定されます。このコマンドは、Sun Fire 880 内蔵記憶装置サブシステムには使用できません。

`probe` [`-p`]
接続された Sun StorEdge A5000 サブシステムおよび各 FC_AL デバイスすべてに関する情報を検索して表示します。この情報には、論理パス名、WWN、および格

納装置名が含まれます。このサブコマンドは、同じ格納装置名を持つ異なる Sun StorEdge A5000 を見つけると、ユーザーに対して警告を発します。

`-p` 物理パス名を表示に含めます。

`qlgc_s_download` [`-f` *fcode-file*]

fcode-file ファイルに格納されている fcode をすべての FC100/P、FC100/2P PCIホストアダプタカードにダウンロードします。このコマンドは対話型で、fcode を各ドライブにダウンロードする前にユーザーに対して確認を促します。`qlgc_s_download` は、シングルユーザーモードでだけ使用してください。入出力操作があるホストアダプタに対して、そのアダプタを更新する目的で `qlgc_s_download` を指定すると、アダプタのリセットの原因となります。新しく更新した fcode は、システムをリブートしたときに実行され、表示できるようになります。

`-f` *fcode-file* `-f` オプションを省略すると、各 FC100/P、FC100/2P PCI カードの、現在のバージョンの fcode が表示されます。

`release` *pathname*

指定されたディスクの予約を解除します。パス名は、ディスクの物理または論理パス名でなければなりません。SPARCstorage Array コントローラのパス名を指定した場合、SPARCstorage Array 中のすべてのディスクの予約が解除されます。

このサブコマンドは、履歴および診断目的以外では使用しないでください。

`remove_device` [`-F`] *enclosure*[,*dev*] . . . | *pathname* . . .

このコマンドを使用して、電源を入れたまま 1 つまたは複数のデバイスを削除することができます。このサブコマンドはすべての格納装置を削除する場合にも指定することができます。このサブコマンドは Sun StorEdge A5000、RSM、および各 FC_AL デバイスに対して指定することができます。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。Sun StorEdge A5000 および各 FC_AL デバイスに対しては、このサブコマンドは、電源を入れたまま 1 つまたは複数のデバイスを削除するすべての過程をユーザーとの対話によって行います。複数のディスクが指定された場合は、それらのディスクが正しいかどうかの確認が行われ、ユーザーは継続するか中止するかを選択することができます。その後で、ディスクまたは格納装置の削除を実行するかどうかの確認が行われ、それらのデバイスの論理パス名が作成および表示されます。

多重ホストディスクの場合、実行する手順は次のとおりです。

- 最初のホストに対して、`luxadm remove_device` コマンドを発行します。続行するかどうかを確認するプロンプトが表示されたら、待機します。
- 2 番目のホストに対して、`luxadm remove_device` コマンドを発行します。続行するかどうかを確認するプロンプトが表示されたら、待機します。
- 最初のホストに対して、`remove_device` コマンドを続行します。デバイスを削除するプロンプトが表示されたら、これを実行します。

- ほかのホストに対して、`luxadm remove_device` コマンドを最後まで実行します。

`-F` 1 つまたは複数のデバイスに対してホットプラグを適用するように `luxadm` に指示します。これらのデバイスがホストで使用されている場合 (*busy* または *reserved* の場合) にも適用されます。ホットプラグオペレーションが強制的に実行されます。

警告: 現在使用されているデータが格納されているデバイスを取り外すと、予期しない結果が発生します。通常は、`-F` を指定しないでホットプラグを適用します。-F は、通常のホットプラグ検査を無効にしたときの結果がわかっているときにだけ使用します。

`reserve` *pathname* 指定されたディスクを、ホストによる排他的利用のために 予約します。パス名は、ディスクの物理または論理パス名でなければなりません。pathname が SPARCstorage Array コントローラのパス名の場合、SPARCstorage Array 中のすべてのディスクが予約されます。

このサブコマンドは、履歴および診断目的以外では使用しないでください。

`set_boot_dev` [`-y`] *pathname* システム PROM 中のブートデバイス変数を、*pathname* で指定された物理デバイス名に設定します。指定する物理デバイス名は、ブロック型特殊デバイス、またはブートファイルシステムのマウント先のディレクトリのパス名です。通常、このコマンドは、PROM の中のデフォルトブートデバイスを設定するためにユーザーに対して確認を促します。`-y` オプションは、ユーザーに対して確認を要求せず、非対話形式で実行するために使用できます。

`start` *pathname* 指定されたSENA のディスクを起動します。

`stop` *pathname...* 指定されたディスク SENA のを停止します。

SENA、Sun Fire 800 内蔵記憶装置サブシステムおよび各 FC_AL デバイスのエキスパートモードサブコマンド

以下のサブコマンドは、経験が豊富なユーザーのみが使用することができます。また、Sun StorEdge A5000 およびファイバチャネルループのみを対象にしています。これらのコマンドは、Sun StorEdge A5000 サブシステムとファイバチャネルループの知識が豊富なユーザーだけが使用することができます。

バスを操作するエキスパートサブコマンドにディスクを指定した場合は、そのサブコマンドは、指定したディスクに接続されたバスを操作します。

`-e bypass [-ab]` *enclosure,dev*
`-e bypass -f` *enclosure*

エンクロージャサービスコントローラに対して、指定したポートおよびデバイスの LRC (ループ冗長回路) をバイパス状態に設定するよう要求します。

このサブコマンドは、以下のオプションをサポートします。

`-a` 指定したデバイスのポート `a` をバイパスします。

`-b` 指定したデバイスのポート `b` をバイパスします。

`-e dump_map` *fibre_channel_HBA_port*

指定したファイバチャネルポート上にあるターゲットデバイスまたはホストバスアダプタの WWN データを表示します。指定したポートにターゲットデバイスが存在しない場合、エラーが返されます。

`-e enable [-ab]` *enclosure,dev*
`-e enable -f` *enclosure*

エンクロージャサービスコントローラに対して、指定したポートおよびデバイスの LRC (ループ冗長回路) を有効な状態に設定するよう要求します。

このサブコマンドは、以下のオプションをサポートします。

`-a` 指定したデバイスのポート `a` を有効にします。

`-b` 指定したデバイスのポート `b` を有効にします。

`-e forcelip` *enclosure*[`,` *dev*] . . . | *pathname* . . .

ループ初期化基本式 (LIP) の処理を使ってリンクを強制的に最初期化します。enclosure または pathname には、ループ上のどのデバイスでも指定することができます。複数のループ構成に対して特定のパスを指定する場合は、パス名を使用します。

このコマンドは経験者向けのコマンドで、使用には注意が必要です。このコマンドは、ループ上のすべてのポートをリセットします。

-e rdls *enclosure*[,*dev*] . . . | *pathname* . . .	enclosure または pathname で指定されたデバイスを含むループにある、すべての使用可能なデバイスのリンクエラーステータス情報を読み込み、表示します。

その他のエキスパートモードサブコマンド

ホットプラグ操作中の制限に関しては「注意事項」を参照してください。以下のサブコマンドは、経験が豊富なユーザーのみが使用することができます。

これらのコマンドは、Sun Fire 880 内蔵記憶装置サブシステムに使用することはできません。

-e bus_getstate *pathname*	指定されたバスの状態を取得、表示します。
-e bus_quiesce *pathname*	指定されたバスを休止します。
-e bus_reset *pathname*	指定されたバスのみをリセットします。
-e bus_resetall *pathname*	指定されたバスおよびすべてのデバイスをリセットします。
-e bus_unquiesce *pathname*	指定されたバスを休止解除します。指定されたデバイス。
-e dev_getstate *pathname*	指定されたデバイスの状態を取得、表示します。
-e dev_reset *pathname*	指定されたデバイスをリセットします。
-e offline *pathname*	指定されたデバイスをオフラインにします。
-e online *pathname*	指定されたデバイスをオンラインにします。

使用例

例 1　システム上にあるすべての Sun StorEdge A5000 および FC_AL の表示

システム上にあるすべての Sun StorEdge A5000 および FC_AL デバイスを検索、表示する例を次に示します。

```
example% luxadm probe
```

例 2　SENA または Sun Fire 880 内蔵記憶装置サブシステムの表示

SENA または Sun Fire 880 内蔵記憶装置サブシステムを表示する例を次に示します。

```
example% luxadm display /dev/es/ses0
```

例 3　2 つのサブシステムの表示

格納装置名を使って 2 つのサブシステムを表示する例を次に示します。

```
example% luxadm display BOB system1
```

例 4　最初のディスクに関する情報の表示

BOB という名前の格納装置の前面にある最初のディスクに関する情報を表示する例を次に示します。前面のディスクを指定する場合は f を指定します。背面のディスクを指定する場合は r を指定します。

```
example% luxadm display BOB,f0
```

例 5　Sun Fire 880 内蔵記憶装置サブシステムに関する情報の表示

Sun Fire 880 内蔵記憶装置サブシステムには、1 セットのディスクしかありません。この場合、そのスロットを指定する場合は *s* を指定します。

```
example% luxadm display BOB,s0
```

例 6　SENA ディスク、格納装置、または各 FC_AL デバイスに関する情報の表示

ポートの WWN が 2200002037001246 の Sun StorEdge A5000 ディスク、格納装置、または各 FC_AL デバイスに関する情報を表示する例を次に示します。

```
example% luxadm display 2200002037001246
```

例 7　サブコマンドとして一意に認識するための文字列

サブコマンドとして一意に認識できるだけの長さの文字列を使用する例を次に示します。

```
example% luxadm disp BOB
```

例 8　エラー情報の表示

格納装置 BOB があるループに関するエラー情報を表示する例を次に示します。

```
example% luxadm display -r BOB
```

例 9　インタフェースボードへの新しいファームウェアのダウンロード

格納装置 BOB のインタフェースボードに新しいファームウェアをダウンロードする例を次に示します (ダウンロードするファイルはデフォルトパスで指定されています)。

```
example% luxadm download -s BOB
```

例 10　SCSI 照会コマンドからの情報の表示

システム上の個々のディスクから SCSI 照会コマンドからの情報を表示する例を次に示します。サブコマンドとして一意に認識できる長さの文字列だけが使用されます。

```
example% luxadm inq /dev/rdsk/c?t?d?s2
```

例 11　ホットプラグによる取り付け

BOB,f1 という名前の格納装置の前面の最初のスロットに新しいドライブをホットプラグで取り付ける例を次に示します。

```
example% luxadm insert_device BOB,f0
```

SF880-1 という名前の Sun Fire 880 内蔵記憶装置サブシステムの最初のスロットに、新しいドライブをホットプラグで取り付ける例を次に示します。

```
example% luxadm insert_device SF880-1,s0
```

例 12　エキスパートサブコマンドの実行

エキスパートサブコマンドを実行する例を次に示します。このサブコマンドは、格納装置 BOB のあるループを強制的に初期化します。

```
example% luxadm -e forcelip BOB
```

例 13　エキスパートモードのホットプラグサブコマンドの使用

エキスパートモードのホットプラグサブコマンドを使用して SSA 上のディスクを削除する例を次に示します。ホットプラグ操作中の制限に関しては「注意事項」を参照してください。

最初の手順では、SCSI デバイスが 2 つ目の SCSI バスによってアクセスされないように、そのデバイスを予約します。

```
example# luxadm reserve /dev/rdsk/c1t8d0s2
```

例 14　ディスクをオフラインにする

次の 2 つの手順では、ディスクをオフラインにし、バスを休止します。

```
example# luxadm -e offline /dev/rdsk/c1t8d0s2
example# luxadm -e bus_quiesce /dev/rdsk/c1t8d0s2
```

例 15　バスの休止解除

ユーザーは、この時点でディスクを取り外し、バスを休止解除し、ディスクをオンラインに戻し、ディスクの予約を解除します。

```
example# luxadm -e bus_unquiesce /dev/rdsk/c1t8d0s2
example# luxadm -e online /dev/rdsk/c1t8d0s2
```

例 15 バスの休止解除 (続き)

```
example# luxadm release /dev/rdsk/c1t8d0s2
```

環境変数

luxadm の実行に影響のある環境変数 LANG に関しては environ(5) を参照してください。

終了ステータス

次の終了値が返されます。

0 正常終了。

–1 エラーが発生しました。

ファイル

- usr/lib/firmware/fc_s/fc_s_fcode
- usr/lib/locale/C/LC_MESSAGES/ibfirmware

属性

属性についての詳細は、attributes(5) を参照してください。

usr/sbin

属性タイプ	属性値
使用条件	system/storage/fc-utilities

関連項目

devlinks(1M), disks(1M), attributes(5), environ(5), ses(7D)

『SAN Configuration and Multipathing Guide』

『Oracle Solaris Cluster 3.3 with Fibre Channel JBOD Storage Device Manual』

注意事項

IEEE 拡張 WWN に関する情報については、『*Tutorial for SCSI use of IEEE Company_ID*』 (R. Snively 著) を参照してください。現在、一部のデバイスドライバのみがホットプラグに対応しています。ホットプラグに対応していないディスクまたはバスにホットプラグが適用されると、メッセージが表示されます。

```
luxadm: can’t acquire "PATHNAME": No such file or directory
```

上記の形式で表示されます。

ルートファイルシステムまたは /usr ファイルシステムを含むバスやスワップデータを含むバスを休止する際は注意してください。そのようなバスを休止すると、デッドロックを引き起こす可能性があり、そのような場合は、システムのリブートが必要となります。

参 照

システム管理コマンド - パート 2

名前

mail.local – store mail in a mailbox

形式

`/usr/lib/mail.local [-f` *sender*`] [-d]` *recipient*

機能説明

`mail.local` reads the standard input up to an end-of-file and appends it to each user's mail file (mailbox). This program is intended to be used by `sendmail(1M)` as a mail delivery agent for local mail. It is not a user interface agent.

Messages are appended to the user's mail file in the `/var/mail` directory. The user must be a valid user name.

Each delivered mail message in the mailbox is preceded by a "Unix From line" with the following format:

`From` *sender_address time_stamp*

The *sender_address* is extracted from the SMTP envelope address (the envelope address is specified with the `-f` option).

A trailing blank line is also added to the end of each message.

The mail files are locked with a `.lock` file while mail is appended.

The mail files are created with mode `660`, owner is set to *recipient*, and group is set to `mail`. If the "biff" service is returned by `getservbyname(3SOCKET)`, the biff server is notified of delivered mail. This program also computes the `Content-Length:` header which will be used by the mailbox reader to mark the message boundary.

オプション

The following options are supported:

`-f` *sender*	Specifies the "envelope from address" of the message. This flag is technically optional, but should be used.
`-d`	Specifies the recipient of the message. This flag is also optional and is supported here for backward compatibility. That is, `mail.local` *recipient* is the same as `mail.local -d` *recipient*.
`-l`	Turn on LMTP mode.
`-r` *from*	Specify the sender's name (for backward compatibility).
`-7`	Do not advertise 8BITMIME support in LMTP mode.
`-b`	Return a permanent error instead of a temporary error if a mailbox exceeds quota.

オペランド

The following operand is supported:

recipient The recipient of the mail message.

環境変数

`TZ` Used to set the appropriate time zone on the timestamp.

終了ステータス

The following exit values are returned:

`0` Successful operation.

`>0` An error occurred.

ファイル

`/tmp/local.`*XXXXXX* temporary files

`/tmp/lochd.`*XXXXXX* temporary files

`/var/mail/`*user_name* user's mail file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

関連項目

`mail(1)`, `comsat(1M)`, `sendmail(1M)`, `getservbyname(3SOCKET)`, `attributes(5)`

名前

makedbm – make a dbm file, or get a text file from a dbm file

形式

```
makedbm [-b] [-l] [-s] [-E] [-i yp_input_file]
      [-o yp_output_name] [-d yp_domain_name]
      [-m yp_master_name] [-S delimiter]
      [-D number_of_delimiters] infile outfile

makedbm [-u dbmfilename]
```

機能説明

The `makedbm` utility takes the *infile* and converts it to a pair of files in `ndbm` format (see `ndbm(3C)`), namely *outfile*`.pag` and *outfile*`.dir`. Each line of the input file is converted to a single `dbm` record. All characters up to the first TAB or SPACE form the key, and the rest of the line is the data. If a line ends with '\' (backslash), the data for that record is continued on to the next line. `makedbm` does not treat '#' (pound-sign) as a special character.

Because `makedbm` is mainly used in generating `dbm` files for the NIS name service, it generates a special entry with the key *yp_last_modified*, which is the date of *infile* (or the current time, if *infile* is '–'). The entries that have keys with the prefix *yp_* are interpreted by NIS server utilities.

オプション

The following options are supported:

`-b`	Insert the `YP_INTERDOMAIN` into the output. This key causes `ypserv(1M)` to use DNS for host name and address lookups for hosts not found in the maps.
`-d` *yp_domain_name*	Create a special entry with the key *yp_domain_name*.
`-D` *number_of_delimiters*	Specify *number_of_delimiters* to skip before forming the key.
`-E`	Delimiters are escaped.
`-i` *yp_input_file*	Create a special entry with the key *yp_input_file*.
`-l`	Lower case. Convert the keys of the given map to lower case, so that, for example, host name matches succeed independent of upper or lower case distinctions.
`-m` *yp_master_name*	Create a special entry with the key *yp_master_name*. If no master host name is specified, *yp_master_name* is set to the local host name.
`-o` *yp_output_name*	Create a special entry with the key *yp_output_name*.
`-s`	Secure map. Accept connections from secure NIS networks only.
`-S` *delimiter*	Specify the *delimiter* to use instead of the default delimiter for forming the key.
`-u` *dbmfilename*	Undo a `dbm` file. Prints out the file in text format, one entry per line, with a single space separating keys from values.

オペランド

The following operands are supported:

infile Input file for `makedbm`. If *infile* is '–' (dash), the standard input is read.

outfile One of two output files in `ndbm` format: *outfile*`.pag` and *outfile*`.dir`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`ypserv(1M)`, `ndbm(3C)`, `attributes(5)`

名前

makemap – create database maps for sendmail

形式

```
makemap [-N] [-d] [-f] [-o] [-r] [-s] [-v] [-C file]
     [-c cachesize] [-D commentchar] [-e] [-l] [-t delim]
     [-u] mantype mapname
```

機能説明

`makemap` creates the database maps used by the keyed map lookups in `sendmail(1M)`. `makemap` reads from the standard input and outputs to the specified *mapname*.

In all cases, `makemap` reads lines from the standard input consisting of two words separated by whitespace. The first is the database key, the second is the value. The value may contain `%`*n* strings to indicated parameter substitution. Literal percents should be doubled (`%%`). Blank lines and lines beginning with `#` are ignored.

`makemap` handles three different database formats. Database format is selected using the *maptype* parameter. See `OPERANDS`.

オプション

The following options are supported:

`-c` *cachesize*	Use the specified hash and B-Tree cache size (*cachesize*).
`-C` *file*	Use the specified sendmail configuration file (*file*) for looking up the TrustedUser option.
`-d`	Allow duplicate keys in the map. This is only allowed on B-Tree format maps. If two identical keys are read, both be inserted into the map.
`-D` *commentchar*	Use the specified character to indicate a comment (which is ignored) instead of the default of '`#`'.
`-e`	Allow empty value (right hand side).
`-f`	Normally, all upper case letters in the key are folded to lower case. This flag disables that behavior. This is intended to mesh with the `-f` flag in the `K` line in `sendmail.cf`. The value is never case folded.
`-l`	List supported map types.
`-N`	Include the null byte that terminates strings in the map. This must match the `-N` flag in the `K` line in `sendmail.cf`
`-o`	Append to an old file. This allows you to augment an existing file.
`-r`	Allow replacement of existing keys. Normally `makemap` complains if you repeat a key, and does not do the insert.
`-s`	Ignore safety checks on maps being created. This includes checking for hard or symbolic links in world writable directories.
`-t` *delim*	Use the specified delimiter (*delim*) instead of whitespace.

`-u` Dump (unmap) the content of the database to standard output. Note that, if the `-t` option is also provided, the specified delimiter is used when the content is dumped instead of whitespace.

`-v` Verbosely print what it is doing.

オペランド

The following operands are supported:

mapname File name of the database map being created.

maptype Specifies the database format. The following *maptype* parameters are available:

- `dbm` Specifies DBM format maps.
- `btree` Specifies B-Tree format maps.
- `hash` Specifies hash format maps.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

関連項目

`editmap(1M)`, `sendmail(1M)`, `attributes(5)`

名前

makeuuid – generate Universal Unique Identifiers

形式

`makeuuid [-e ether] [-n count] [-R root]`

機能説明

The `makeuuid` command generates UUIDs (Universal Unique Identifiers) conforming to the OSF DCE specification for UUIDs. The specification states:

"A UUID is an identifier that is unique across both space and time, with respect to the space of all UUIDs. A UUID can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects across a network.

"The generation of UUIDs does not require a registration authority for each single identifier. Instead, it requires a unique value over space for each UUID generator. This spatially unique value is [normally] specified as an IEEE 802 address, which is usually already applied to network-connected systems."

The `makeuuid` command generates one or more UUIDs on the standard output.

オプション

The `makeuuid` command supports the following options:

`-e` *ether*
: Supplies an alternate address to be used in the generation of the UUIDs. Normally, the system's Ethernet address is acquired and used during the generation of a UUID. However, this requires root privileges to open and read the network devices. If this is not possible, you must supply an alternate Ethernet address.

`-n` *count*
: Generate multiple UUIDs. This option generates the specified number of UUIDs, one per line. Using this form is more efficient than, and functionally equivalent to, calling the `makeuuid` command multiple times. This can be used, for example, when a large number of UUIDs need to be generated for a given application.

`-R` *root*
: Use *root* as the root filesystem path when updating the shared state file (see FILES). The shared state file must be writable by the user running `makeuuid`, otherwise no UUIDs will be generated and the command will return in failure.

  注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

使用法

Normally, you run the `makeuuid` command with root privileges, as the Ethernet address and state files can be easily accessed and updated. If this is not possible, you must use the `-R` and `-e` options to specify an alternate root and Ethernet address to use when calculating the UUIDs.

使用例

例 1 Generating Multiple UUIDs

The following command generates 3000 UUIDs:

```
example# makeuuid -n 3000
```

例 2 Invoking Without Root Privileges

If you cannot obtain root privileges, you must specify an alternate Ethernet address and state file location:

```
example% makeuuid -e 11:22:33:44:55:66 -R /export/root/example2
```

See the caveat on the use of the `-R` option in the description of that option, above.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Out of memory.

`–1` Invalid Ethernet address given or access denied.

ファイル

`/var/sadm/system/uuid_state` UUID state file. Use of time values is one way that UUID generators, such as `makeuuid`, guarantee uniqueness. A state file is a mechanism that allows `makeuuid` to "remember" the last time value it used so it can increment that value for use in a new UUID. See the Internet Draft UUIDs and GUIDs, dated February 4, 1998, for details on the state file mechanism.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/product-registry

関連項目

`Intro(3)`, `libwsreg`(3LIB), `attributes(5)`

注意事項

The formal UUID specification is in the OSF DCE specification, available at `www.opengroup.org`. As of the date of publication of this man page, a copy of the specification is available at:

```
http://www.opengroup.org/onlinepubs/9629399/apdxa.htm
```

Sun has no control over the availability of documents on the `www.opengroup.org` web site.

名前

masfcnv – SNMP configuration migration script

形式

```
/usr/lib/sma_snmp/masfcnv [-cimnrs] [-l agentmaster]
     [-p enabledisableerror] [-t noneadd]
     [-u agentmastererror] [-y agentmastererror]

masfcnv [-V]

masfcnv [-?]
```

機能説明

The `masfcnv` script is used to assist the system administrator in migrating an existing set of configuration files for the Sun SNMP Management Agent for Sun Fire and Netra Systems (MASF) to the Systems Management Agent (SMA).

The script accepts as input the currently installed set of MASF and SMA configuration files and outputs a new set of SMA configuration files. Existing SMA configuration files are backed up by appending `.bak` to the filename. The administrator can choose to output the new configuration to standard output, instead of replacing the current configuration, by specifying the `-n` option.

The migration script must be run as the superuser. Failure to do so causes the script to exit with an error message. Before running the script you should ensure that both the SMA and MASF agents are not running. If the agents are running they will be shut down by the script.

The migration script installs a new startup script for the MASF agent in `/etc/init.d`, as well as a backup of the old script. During migration, MASF will be configured as an AgentX subagent of SMA. All migration settings will be migrated to the SMA configuration file.

The migration script aborts if any unrecognized directives are found in either the MASF configuration files or the SMA configuration files. This can be overridden with the `-i` option. If this option is selected, the behavior is to retain unrecognized directives that were present in the SMA configuration, but remove those present in the MASF configuration.

The migration script then proceeds to migrate access control and trap configuration. As a side effect of running the migration script, the following directives might be expanded by the script into multiple directives with an equivalent interpretation:

- `rwcommunity`
- `rocommunity`
- `rwuser`
- `rouser`
- `trapcommunity`
- `trapsink`
- `trap2sink`
- `informsink`

Access Control Migration

Access control directives are expanded into the equivalent com2sec, group, access and view directives. Existing group names are renamed by prepending a prefix to avoid conflict with any which may already be defined in SMA.

When migrating SNMPv1 or v2c access control, a conflict can occur if both MASF and SMA configuration files have defined access permissions for the same community and source address. The default behavior is to abort with a message, unless a use of the `-y` option specifies otherwise. If `-y agent` is specified then the MASF configuration takes precedence. If `-y master` is specified then the SMA configuration is retained.

When migrating USM configuration (SNMPv3), a conflict can occur if both SMA and MASF configurations define a user with the same `securityName`. If this occurs, the behavior of the script is determined by the `-u` option. If `-u agent` is specified, the configuration of the user defined in the MASF configuration files is the one that is retained. Otherwise, if the `-u master` option is specified, the use defined in the SMA configuration files is retained.

By default, the migration script attempts to migrate USM users from MASF to SMA. The script determines whether there are any SNMPv3 users present in the SMA configuration and whether the default `engineID` has been overridden in the SMA configuration files. If neither of these conditions obtain, then the any `usmUser` statements containing localized authentication keys can be migrated to SMA, along with the MASF `engineID`. This results in the `engineID` of the SMA master agent changing.

If the script determines that there are existing SNMPv3 users or a manually configured `engineID` present in the SMA configuration, only those users defined in `createUser` statements are transferred. Those users that were defined in `usmUser` statements are transferred but will have their passwords reset to a random value. You should notify your users of their new password or reset the password yourself by editing the newly-generated configuration file.

Trap/Inform Migration

The migration script performs a check to determine whether a trap destination defined for MASF is already specified in an existing SMA `trapsink`, `trap2sink` or `informsink` directive. If this is the case, then the directive in the MASF configuration will be discarded to avoid duplicate traps/informs being received.

`trapsink`, `trap2sink` and `informsink` directives specified in the existing SMA configuration are considered valid destinations for MASF traps/informs and will receive them from the MASF subagent after migration.

If the `-t none` option was specified on the command line, the migration script carries over any remaining MASF trap/inform directives without modification.

If the `-t add` option was specified (the default), the migration script expands any `trapsink`, `trap2sink`, or `informsink` directives to use the `TARGET-MIB` and `NOTIFICATION-MIB`. The `TARGET-MIB` specifies targets using IP addresses, so it might be desirable to use the `-t none` option if, for example, the network allocates IP addresses to hostnames dynamically by means of DHCP.

The expanded directives defines filters specific to the MASF agent so that traps from other subagents will not be received by migrated trap destinations. Existing filters present in the SMA configuration are, by default, not modified and might or might not receive MASF traps, depending upon the filters that were originally defined for them.

If the `-l` option is specified, any filters already defined in the `TARGET-MIB` and the `NOTIFICATION-MIB` for SMA are extended to include traps from MASF. In the event that a trap destination is already configured in the `TARGET-MIB` with the same target address and community as an existing MASF trap/inform sink, a conflict will arise.

If `-l agent` was specified and a conflict arises, the migration script uses the target SNMP parameters (that is, the SNMP version and choice of trap/inform) defined by the MASF `trap/informsink` directive to send traps to this destination. Otherwise, if the `-l master` option was specified, the conflict will be resolved using the target SNMP parameters specified in the SMA configuration.

Miscellaneous

If the migration script encounters in the MASF configuration file any of the directives listed below and the directives are either not present or differ from the SMA configuration, the script will log a warning message.

- `syslocation`
- `syscontact`
- `sysname`
- `sysservices`
- `agentgroup`
- `agentuser`
- `authtrapenable`

オプション

The following options are supported:

`-?` `--help`	Displays usage information.
`-c` `--no-community`	Do not transfer v1/v2c communities.
`-i` `--ignore-unrecognized-directives`	Continue processing if unrecognized directives are present.
`-l agent|master` `--master-trap-target=agent|master`	If `agent` is specified, the existing SMA trap targets will be configured to receive traps that were previously sent to destinations for the Sun Fire SNMP agent. If `master` is specified, the targets will be configured to receive Sun Fire SNMP traps, but existing SNMP target parameters will be used.

`-m` `--no-usmuser`	Do not transfer usm (v3) users.
`-n` `--dry-run`	Run the migration without modifying any files. If an error arises, continue processing. This can be used to determine the likely migration issues.
`-p enable|disable|error` `--use-agent-port=enable|disable|error`	Indicates whether the port originally used by the Sun Fire SNMP agent should be used by the SMA agent after migration (if the two agents are using different ports). If `enable` is specified, then the port used by the Sun Fire SNMP agent will also be used by the SMA agent after migration. If `disable` is specified, the ports used by SMA will not be updated by the migration tool. If the `error` option is specified and the SMA agent is not already using the same ports as those used by the original Sun Fire SNMP agent, an error is reported and the migration process is terminated. If no option is specified the default behavior is equivalent to the `error` flag.
`-r` `--no-trap`	Do not transfer trap destinations.
`-s` `--skip-user`	If a user is found in the MASF configuration file that cannot be created in the new configuration because of a change in the engine ID, then output a message indicating that the user could not be migrated (needs to be manually recreated) and continue processing. If this option is not present, the migration tool will consider such a situation as an error and abort.
`-t none|add` `--trap-filter=none|add`	If `none` is specified then the script will copy trap directives directly. The administrator might need to manually update the

configuration file to ensure traps are only delivered to their intended destinations. If `add` is specifed, trap filters will be constructed so that traps originating from the original Sun Fire SNMP agent are delivered only to the destinations that originally received them. The default behavior is `add`.

`-u agent | master | error`
`--select-user=agent | master | error`

Specifies that if a user with the same name is found in both configuration files that the conflict is to be resolved using the specified configuration file as input. Selecting a user from a particular file will also cause the group declaration for that user to be taken from the same file. If `agent` is specified then the user will be taken from the configuration file for the Sun Fire SNMP Agent. If `master` is specified, the user will be taken from the SMA configuration. Otherwise, if `error` is given, the script will terminate. If this option is not present, the default behavior is equivalent to the `error` flag.

`-V`
`--version`

Display the version of this script.

`-y agent | master | error`
`--select-community=agent | master | error`

Specifies that if a community with the same name is found in both configuration files that the conflict is to be resolved using the specified configuration file as input. If `agent` is specified then the community will be taken from the configuration file for the Sun Fire SNMP Agent. If `master` is specified, the community will be taken from the SMA configuration. Otherwise, if `error` is given, the script will terminate. If this option is not present, the default behavior is equivalent to the `error` flag.

使用例

例 1 Simplest Case

The command shown below is appropriate for a simple migration. The migration fails if there are any potential conflicts.

```
# masfcnv
```

例 2 Migrating Such That MASF Settings Override

To migrate the MASF configuration such that it will always succeed, that MASF settings will override in the event of a conflict with SMA, and that access will still be provided on the original MASF port, enter:

```
# masfcnv -is -l agent -p enable -u agent -y agent
```

例 3 Dry Run, Retaining SMA Settings

To attempt a dry run and migrate the configuration such that any conflicts will be resolved by retaining existing SMA settings, enter:

```
masfcnv -l master -u master -y master
```

終了ステータス

`0` Success.

`non-zero` A problem occurred during migration.

ファイル

`/etc/sma/snmp/snmpd.conf`
`/var/sma_snmp/snmpd.conf` SMA configuration files

`/etc/opt/SUNWmasf/conf/snmpd.conf`
`/var/opt/SUNWmasf/snmpd.dat` MASF configuration files

`/tmp/sma_migration.log` `masfcnv` log file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsmcmd
Interface Stability	Committed

関連項目

`attributes(5)`

注意事項

The former path to this utility, `/usr/sfw/lib`, is now a link to `/usr/lib`.

名前

mdlogd – Solaris Volume Manager daemon

形式

`mdlogd`

機能説明

`mdlogd` implements a simple daemon that watches the system console looking for messages written by the Solaris Volume Manger. When a Solaris Volume Manager message is detected, `mdlogd` sends a generic SNMP trap.

To enable traps, you must configure `mdlogd` into the SNMP framework. See 『Solaris ボリュームマネージャの管理』.

使用法

`mdlogd` implements the following SNMP MIB:

```
SOLARIS-VOLUME-MGR-MIB DEFINITIONS ::= BEGIN
        IMPORTS
                 enterprises FROM RFC1155-SMI
                 DisplayString FROM SNMPv2-TC;

        -- Sun Private MIB for Solaris Volume Manager


        sun       OBJECT IDENTIFIER ::= { enterprises 42 }
        sunSVM       OBJECT IDENTIFIER ::= { sun 104 }

        -- this is actually just the string from /dev/log that
        -- matches the md: regular expressions.
        -- This is an interim SNMP trap generator to provide
        -- information until a more complete version is available.

        -- this definition is a formalization of the old
        -- Solaris DiskSuite mdlogd trap mib.

        svmOldTrapString OBJECT-TYPE
                        SYNTAX DisplayString (SIZE (0..255))
                        ACCESS read-only
                        STATUS mandatory
                        DESCRIPTION
                        "This is the matched string that
                         was obtained from /dev/log."
        ::= { sunSVM 1 }

        -- SVM Compatibility ( error trap )

        svmNotice        TrapTRAP-TYPE
                        ENTERPRISE sunSVM
                        VARIABLES { svmOldTrapString }
                        DESCRIPTION
                                "SVM error log trap for NOTICE.
                                 This matches 'NOTICE: md:'"
```

```
        ::= 1

        svmWarningTrap  TRAP-TYPE
                        ENTERPRISE sunSVM
                        VARIABLES { svmOldTrapString }
                        DESCRIPTION
                                "SVM error log trap for WARNING..
                                 This matches 'WARNING: md:'"
        ::= 2

        svmPanicTrap    TRAP-TYPE
                        ENTERPRISE sunSVM
                        VARIABLES { svmOldTrapString }
                        DESCRIPTION
                                "SVM error log traps for PANIC..
                                This matches 'PANIC: md:'"
        ::= 3
END
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWlvma, SUNWlvmr
Interface Stability	Obsolete

関連項目

snmpdx(1M), attributes(5)

『Solaris ボリュームマネージャの管理』

名前

mdmonitord – daemon to monitor metadevices

形式

`/usr/sbin/mdmonitord [-t time_interval]`

機能説明

The `mdmonitord` utility is part of Solaris Volume Manager. It monitors and checks RAID1 (mirrors), RAID5 and hot spares.

There are two methods for checking:

- At fixed time intervals.
- When a RAID-1 (mirror), RAID-5, or hot spare fails. A failure generates an error event which triggers a check of these metadevices.

オプション

The following options are supported:

`-t` Time interval in seconds. The default value is 0, which causes probes to occur only upon an error. If you want to run `mdmonitord` at a regular interval, a value of 1800 (seconds, every half hour) is recommended as a starting point.

終了ステータス

The following exit values are returned:

`0` Successful completion.

>0 An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm

関連項目

`svcs(1)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metarename(1M)`, `metareplace(1M)`, `metaset(1M)`, `metassist(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `svcadm(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `smf(5)`, `md(7D)`

『Solaris ボリュームマネージャの管理』

注意事項

Since frequent probes can affect performance, it is recommended that the intervals between probes be limited.

The `mdmonitord` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/mdmonitor
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

mdnsd – マルチキャスト DNS デーモン

形式

```
mdnsd [-debug]
```

機能説明

mdnsd は、マルチキャスト DNS および DNS サービス検出用のデーモンプログラムです。

mdnsd デーモンは、UDP ポート 5353 上でマルチキャスト DNS クエリーパケットを待機します。mdnsd は、応答できるクエリーを受信すると、適切なマルチキャスト DNS 応答パケットを発行します。

さらに、mdnsd デーモンは、クライアントプロセスに代わってマルチキャスト DNS クエリーを実行し、応答のキャッシュを維持します。

mdnsd デーモンには、ユーザーが指定可能なコマンド行引数はありません。ユーザーは mdnsd を手動で実行しないでください。mdnsd デーモンは Solaris サービス管理機能 (SMF) により管理され、その操作は svcadm(1M) コマンドにより、次の障害管理リソース識別子 (FMRI) に基づいて行われます。

```
svc:/network/dns/multicast:default
```

mdnsd の内部状態をデバッグおよび診断目的で調べるには、これに SIGUSR1 信号を送信します。これで、syslog(3C) 機能により、内部状態のスナップショットの要約が記録されます。mdnsd は、syslog 機能コードデーモンおよび情報優先度を使用します。

オプション

mdnsd デーモンは次のオプションを認識します。

-debug　デバッグモード。mdnsd デーモンは、出力を標準エラーに送信します。バックグラウンドでは実行されません。このオプションは、デーモンのデバッグ時にのみ使用します。

構成

ネームサービススイッチの構成ファイル nsswitch.conf(4) 内で mdns をホストやipnode のホストとして指定することにより、マルチキャスト DNS を使ってホスト名やホストアドレスを検索できます。マルチキャスト DNS を使用してホスト名やホストアドレスを照会する際の構成オプションは、SMF リポジトリに格納されます。この構成を変更するには、次の障害管理リソース識別子 (FMRI) を使用して svccfg(1M) コマンドを実行します。

```
svc:/network/dns/multicast:default
```

マルチキャスト DNS を使用してホスト名やホストアドレスを照会する際の構成オプションは、「nss_mdns_config」という名前のプロパティーグループに格納されます。この構成オプションを構成するプロパティーは次のとおりです。

search　ホスト名検索に使用する検索ドメインのリスト。デフォルトでは、検索ドメインは構成に含まれていません。現在のところ、検索リストは 6 つのドメインに制限されています。

valid　有効なドメインのリスト。ドメインの有効性の確認は、マルチキャスト DNS を使ってドメインのホスト名やホストアドレスを検索する前に行われます。検索リストに指定されたドメインは、常に有効なリストに含まれています。現在のところ、このリストに含めることができる有効なドメインの数は 10 個に制限されています。

これらの構成オプションは「流動的」であり、将来のリリースで変更される可能性があります。

ファイル

/usr/lib/inet/mdnsd

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/mdns
インタフェースの安定性	流動的

関連項目

dns-sd(1M), svcadm(1M), svccfg(1M), syslog(3C),nsswitch.conf(4), attributes(5)

マルチキャスト DNS については、http://www.multicastdns.org/ を参照してください。

DNS サービス検出については、http://www.dns-sd.org/ を参照してください。

名前

medstat – check the status of mediator hosts for a given diskset

形式

`/usr/sbin/medstat [-q] -s` *setname*

機能説明

If a specified diskset has been configured for mediators, `medstat` attempts to contact these hosts to see if they are accessible and returns the results of the communication.

オプション

`-q` This optional argument disables the printing of informative text. When used with `-q`, `medstat` still prints error messages and returns a result code.

`-s` *setname* Specifies the name of a diskset on which `medstat` will work.

使用例

例 1 Checking diskset

This example checks the mediator hosts for the selected diskset.

```
# medstat -s relo-red
```

The name of the diskset is `relo-red`. The `medstat` command prints the status for each mediator host. Additionally, if the mediator quorum is met, either through a “golden" mediator host or because half+1 of the mediator hosts respond, the exit code is 0. If the quorum is not met, then the exit code is 1. If no mediator hosts have been configured for the named diskset, the exit code is 2. The status field will contain one of the following values: Unreachable, Bad, Fatal, or Ok, where Unreachable indicates an RPC/communication problem, Bad indicates an error in the mediator data, Fatal indicates any other error condition, and Ok indicates no error conditions.

ファイル

`/etc/lvm/meddb` Contains the mediator data for a host that has been selected as a mediator host.

終了ステータス

The following exit values are returned:

`0` Successful completion.

>0 An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metareplace(1M)`, `metaset(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.cf(4)`, `md.tab(4)`, `mddb.cf(4)`, `meddb(4)`, `mediator(7D)`

Sun Cluster documentation, Solaris Volume Manager Administration Guide

注意事項 This command is designed for use in the high availability product.

名前

metaclear – delete active metadevices and hot spare pools

形式

```
/usr/sbin/metaclear -h
/usr/sbin/metaclear [-s setname] -a [-f]
/usr/sbin/metaclear component
/usr/sbin/metaclear [-s setname] [-f] metadevice... hot_spare_pool...
/usr/sbin/metaclear [-s setname] -r [-f] metadevice... hot_spare_pool...
/usr/sbin/metaclear [-s setname] -p component
/usr/sbin/metaclear [-s setname] -p metadevice
```

機能説明

The `metaclear` command deletes the specified metadevice or *hot_spare_pool*, or purges all soft partitions from the designated component. Once a metadevice or hot spare pool is deleted, it must be re-created using `metainit` before it can be used again.

Any metadevice currently in use (open) cannot be deleted.

オプション

Root privileges are required for all of the following options except `-h`.

`-a`	Deletes all metadevices and configured hot spare pools in the set named by `-s`, or the local set by default.
`-f`	Deletes (forcibly) a metadevice that contains a subcomponent in an error state.
`-h`	Displays usage message.
`-p`	Deletes (purges) all soft partitions from the specified metadevice or component.
`-r`	Recursively deletes specified metadevices and hot spare pools, but does not delete metadevices on which others depend.
`-s` *setname*	Specifies the name of the diskset on which `metaclear` will work. Using the `-s` option causes the command to perform its administrative function within the specified diskset. Without this option, the command performs its function on local metadevices and/or hot spare pools.

オペランド

metadevice ...	Specifies the name(s) of the metadevice(s) to be deleted.
component	Specifies the `c*d*t*s*` name(s) of the components containing soft partitions to be deleted.
hot_spare_pool ...	Specifies the name(s) of the hot spare pools to be deleted. Names for hot spare pools can be any legal file name that is composed of alphanumeric characters, a dash (“-”), an underscore (“_”), or a period (“.”). Names must begin with a letter. The words “all” and “none“ are reserved and cannot be used.

使用例

例 1 Deleting Various Devices

The following example deletes a metadevice named d10.

```
# metaclear /dev/md/dsk/d10
```

The following example deletes all local metadevices and hot spare pools on the system.

```
# metaclear -a
```

The following example deletes a mirror, mymirror, with a submirror in an error state.

```
# metaclear -f mymirror
```

The following example deletes a hot spare pool, hsp001.

```
# metaclear hsp001
```

The following example deletes a soft partition, d23.

```
# metaclear d23
```

The following example purges all soft partitions on the slice c2t3d5s2 if those partitions are not being used by other metadevices or are not open.

```
# metaclear -p c2t3d5s2
```

The following example purges soft partitions from a metadevice.

```
# metaclear -p d2
d3: Soft Partition is cleared
d4: Soft Partition is cleared
d5: Soft Partition is cleared
```

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

名前

metadb – create and delete replicas of the metadevice state database

形式

```
/usr/sbin/metadb -h

/usr/sbin/metadb [-s setname]

/usr/sbin/metadb [-s setname] -a [-f] [-k system-file] mddbnn

/usr/sbin/metadb [-s setname] -a [-f] [-k system-file]
     [-c number] [-l length] slice...

/usr/sbin/metadb [-s setname] -d [-f] [-k system-file] mddbnn

/usr/sbin/metadb [-s setname] -d [-f] [-k system-file] slice...

/usr/sbin/metadb [-s setname] -i

/usr/sbin/metadb [-s setname] -p [-k system-file]
     [mddb.cf-file]
```

機能説明

The `metadb` command creates and deletes replicas of the metadevice state database. State database replicas can be created on dedicated slices, or on slices that will later become part of a simple metadevice (concatenation or stripe) or RAID5 metadevice. Do not place state database replicas on fabric-attached storage, SANs, or other storage that is not directly attached to the system and available at the same point in the boot process as traditional SCSI or IDE drives. See `NOTES`.

The metadevice state database contains the configuration of all metadevices and hot spare pools in the system. Additionally, the metadevice state database keeps track of the current state of metadevices and hot spare pools, and their components. Solaris Volume Manager automatically updates the metadevice state database when a configuration or state change occurs. A submirror failure is an example of a state change. Creating a new metadevice is an example of a configuration change.

The metadevice state database is actually a collection of multiple, replicated database copies. Each copy, referred to as a replica, is subject to strict consistency checking to ensure correctness.

Replicated databases have an inherent problem in determining which database has valid and correct data. To solve this problem, Volume Manager uses a *majority consensus algorithm*. This algorithm requires that a majority of the database replicas be available before any of them are declared valid. This algorithm strongly encourages the presence of at least three initial replicas, which you create. A consensus can then be reached as long as at least two of the three replicas are available. If there is only one replica and the system crashes, it is possible that all metadevice configuration data can be lost.

The majority consensus algorithm is conservative in the sense that it will fail if a majority consensus cannot be reached, even if one replica actually does contain the most up-to-date data. This approach guarantees that stale data will not be accidentally used, regardless of the failure scenario. The majority consensus algorithm accounts for the following: the system will

stay running with exactly half or more replicas; the system will panic when less than half the replicas are available; the system will not reboot without one more than half the total replicas.

When used with no options, the `metadb` command gives a short form of the status of the metadevice state database. Use `metadb -i` for an explanation of the flags field in the output.

The initial state database is created using the `metadb` command with both the `-a` and `-f` options, followed by the slice where the replica is to reside. The `-a` option specifies that a replica (in this case, the initial) state database should be created. The `-f` option forces the creation to occur, even though a state database does not exist. (The `-a` and `-f` options should be used together only when no state databases exist.)

Additional replicas beyond those initially created can be added to the system. They contain the same information as the existing replicas, and help to prevent the loss of the configuration information. Loss of the configuration makes operation of the metadevices impossible. To create additional replicas, use the `metadb -a` command, followed by the name of the new slice(s) where the replicas will reside. All replicas that are located on the same slice must be created at the same time.

To delete all replicas that are located on the same slice, the `metadb -d` command is used, followed by the slice name.

When used with the `-i` option, `metadb` displays the status of the metadevice state databases. The status can change if a hardware failure occurs or when state databases have been added or deleted.

To fix a replica in an error state, delete the replica and add it back again.

The metadevice state database (`mddb`) also contains a list of the replica locations for this set (local or shared diskset).

The local set `mddb` can also contain host and drive information for each of the shared disksets of which this node is a member. Other than the diskset host and drive information stored in the local set `mddb`, the local and shared diskset `mddbs` are functionality identical.

The `mddbs` are written to during the resync of a mirror or during a component failure or configuration change. A configuration change or failure can also occur on a single replica (removal of a `mddb` or a failed disk) and this causes the other replicas to be updated with this failure information.

オプション

Root privileges are required for all of the following options except `-h` and `-i`.

The following options can be used with the `metadb` command. Not all the options are compatible on the same command line. Refer to the `SYNOPSIS` to see the supported use of the options.

`-a`
Attach a new database device. The `/kernel/drv/md.conf` file is automatically updated with the new information and the `/etc/lvm/mddb.cf` file is updated as well. An alternate way to create replicas is by defining them in the `/etc/lvm/md.tab` file and specifying the assigned name at the command line in the form, `mddb`*nn*, where *nn* is a two-digit number given to the replica definitions. Refer to the `md.tab(4)` man page for instructions on setting up replicas in that file.

`-c` *number*
Specifies the number of replicas to be placed on each device. The default number of replicas is 1.

`-d`
Deletes all replicas that are located on the specified *slice*. The `/kernel/drv/md.conf` file is automatically updated with the new information and the `/etc/lvm/mddb.cf` file is updated as well.

`-f`
The `-f` option is used to create the initial state database. It is also used to force the deletion of replicas below the minimum of one. (The `-a` and `-f` options should be used together only when no state databases exist.)

`-h`
Displays a usage message.

`-i`
Inquire about the status of the replicas. The output of the `-i` option includes characters in front of the device name that represent the status of the state database. Explanations of the characters are displayed following the replica status and are as follows:

`d`
replica does not have an associated device ID.

`o`
replica active prior to last `mddb` configuration change

`u`
replica is up to date

`l`
locator for this replica was read successfully

`c`
replica's location was in `/etc/lvm/mddb.cf`

`p`
replica's location was patched in kernel

`m`
replica is master, this is replica selected as input

`r`
: replica does not have device relocation information

`t`
: tagged data is associated with the replica

`W`
: replica has device write errors

`a`
: replica is active, commits are occurring to this

`M`
: replica had problem with master blocks

`D`
: replica had problem with data blocks

`F`
: replica had format problems

`S`
: replica is too small to hold current database

`R`
: replica had device read errors

`B`
: tagged data associated with the replica is not valid

`-k` *system-file*
: Specifies the name of the kernel file where the replica information should be written. The default *system-file* is `/kernel/drv/md.conf`. This option is for use with the local diskset only.

`-l` *length*
: Specifies the size of each replica. The default *length* is 8192 blocks, which should be appropriate for most configurations. “Replica” sizes of less than 128 blocks are not recommended.

`-p`
: Specifies updating the system file (`md.conf`) in the current working directory with entries from the `/etc/lvm/mddb.cf` file. This option is normally used to update a newly built system before it is booted for the first time. If the system has been built on a system other than the one where it will run, the location of the `mddb.cf` on the local machine can be passed as an argument. The system file to be updated can be changed using the `-k` option. This option is for use with the local diskset only.

`-s` *setname*
: Specifies the name of the diskset on which the `metadb` command will work. Using the `-s` option will cause the command to perform its administrative function within the specified diskset. Without this option, the command will perform its function on local database replicas.

slice
: Specifies the logical name of the physical slice (partition), such as `/dev/dsk/c0t0d0s3`.

使用例

例 1 Creating Initial State Database Replicas

The following example creates the initial state database replicas on a new system.

```
# metadb -a -f c0t0d0s7 c0t1d0s3 c1t0d0s7 c1t1d0s3
```

The `-a` and `-f` options force the creation of the initial database and replicas. You could then create metadevices with these same slices, making efficient use of the system.

例 2 Adding Two Replicas on Two New Disks

This example shows how to add two replicas on two new disks that have been connected to a system currently running Volume Manager.

```
# metadb -a c0t2d0s3 c1t1d0s3
```

例 3 Deleting Two Replicas

This example shows how to delete two replicas from the system. Assume that replicas have been set up on `/dev/dsk/c0t2d0s3` and `/dev/dsk/c1t1d0s3`.

```
# metadb -d c0t2d0s3 c1t1d0s3
```

Although you can delete all replicas, you should never do so while metadevices still exist. Removing all replicas causes existing metadevices to become inoperable.

ファイル

`/etc/lvm/mddb.cf`
: Contains the location of each copy of the metadevice state database.

`/etc/lvm/md.tab`
: Workspace file for metadevice database configuration.

`/kernel/drv/md.conf`
: Contains database replica information for all metadevices on a system. Also contains Solaris Volume Manager configuration information.

終了ステータス

The following exit values are returned:

`0`
: successful completion

`>0`
: an error occurred

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

注意事項

Replicas cannot be stored on fabric-attached storage, SANs, or other storage that is not directly attached to the system. Replicas must be on storage that is available at the same point in the boot process as traditional SCSI or IDE drives. A replica can be stored on a:

- Dedicated local disk partition
- Local partition that will be part of a volume
- Local partition that will be part of a UFS logging device

名前

metadevadm – update metadevice information

形式

```
/usr/sbin/metadevadm [-h] [-n] [ [-l]-r] [-s setname]
     [-u disk_specifier] [-v]
```

機能説明

The `metadevadm` command facilitates the administration of device ID entries in Solaris Volume Manager. Use this command when the pathname stored in the metadevice state database no longer correctly addresses the device or when a disk drive has had its device ID changed.

This command requires root privileges.

オプション

The following options are supported.

`-h`	Provide a help display.
`-l`	Specify that `metadevadm` log to `syslog(3C)`. `metadevadm` logs to the `DAEMON` facility at the `ERR` level by default. See `syslog.conf(4)` for additional information on changing logging levels. Use this option anytime. It is most useful in startup scripts and less useful interactively.
`-n`	Emulate the effect of a command, without making any changes to the system.
`-r`	Recompute the pathname and disk specifier (including slice) associated with all devices in the metadevice state database if the device supports device IDs. If a device does not support device IDs or the device is not available, then no action is taken for that device. Use this option when the disk has been moved or readdressed. This option is run automatically at boot time to detect device ID changes and update the state database.
`-s` *setname*	Specify the name of the disk set on which `metadevadm` works. This option causes the command to perform its administrative function within the specified disk set. Without this option, the command performs its function on devices in the local disk set.
`-u` *disk_specifier*	Obtain the device ID associated with the *disk_specifier* (for example, `c1t2d0`) of a device and update the metadevice state database. If the device ID has not changed this option does nothing. Use this option when a disk drive has had its device ID changed during a firmware upgrade or due to changing the controller of a storage subsystem.
`-v`	Execute in verbose mode. This option has no effect when used with `-u`. Verbose is the default.

使用例

例 1 Updating Device ID of Disk

The following example updates the device `c2t3d0`:

```
# metadevadm -u c2t3d0
Updating SLVM device relocation information for c2t3d0.
Old device reloc information: id1928019239129312301201201001201209 1398
New device reloc information: id1938019239129312301201201001201209 1398
```

The following example is a variation of the preceding, using the full pathname.

```
# metadevadm -u /dev/dsk/c2t3d0
```

The following example uses the `-n` option, which means that the command is emulated, but does not take effect. Note that when the `-v` option is used with `-u`, `-v` has no effect (verbose is the default).

```
# metadevadm -u -v -n c2t3d0
Updating SLVM device relocation information for c2t3d0.
Old device reloc information: id1928019239129312301201201001201209 1398
New device reloc information: id1938019239129312301201201001201209 1398
```

例 2 Recomputing Pathnames

In the following example, all device names are valid.

```
# metadevadm -r
Disk movement detected.
Updating device names in SLVM.
```

In the following example, once again device names are valid.

```
# metadevadm -r -v
Disk movement detected.
Updating device names in SLVM.
c0t0d0s0 changed to c0t0d1s0 from device relocation information
id12098123lkmklsdjaasdkfjadfjakds
```

In the following example, `metadevadm` detects an invalid device name.

```
# metadevadm -r
Invalid device relocation information detected in SLVM.
Please check status of following disk(s):
c3t0d0
```

戻り値

The following exit values are returned:

`0` Command was successful.

`1` `metadevadm` encountered an error condition.

2 An invalid device ID was detected when using the -r option. This is for use in the rc2.d script. See init.d(4).

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

名前

metahs – manage hot spares and hot spare pools

形式

```
/usr/sbin/metahs [-s setname] -a all component
/usr/sbin/metahs [-s setname] -a hot_spare_pool [component]
/usr/sbin/metahs [-s setname] -d hot_spare_pool [component]
/usr/sbin/metahs [-s setname] -d all component
/usr/sbin/metahs [-s setname] -e component
/usr/sbin/metahs [-s setname] -r hot_spare_pool component-old
/usr/sbin/metahs [-s setname] -r all component-old component-new
/usr/sbin/metahs [-s setname] -i [hot_spare_pool]...
```

機能説明

The `metahs` command manages existing hot spares and hot spare pools. It is used to add, delete, enable, and replace components (slices) in hot spare pools. Like the `metainit` command, the `metahs` command can also create an initial hot spare pool. The `metahs` command does not replace a component of a metadevice. This function is performed by the `metareplace` command.

Hot spares are always in one of three states: available, in-use, or broken. Available hot spares are running and ready to accept data, but are not currently being written to or read from. In-use hot spares are currently being written to and read from. Broken hot spares are out of service and should be repaired. The status of hot spares is displayed when `metahs` is invoked with the `-i` option.

Solaris Volume Manager supports storage devices and logical volumes, including hot spares, greater than 1 terabyte (TB) when Solaris 10 is running a 64-bit kernel.

If a system with large volumes or hot spares is rebooted under a 32-bit Solaris 10 kernel, the large volumes are visible through `metastat` output, but they cannot be accessed, modified or deleted, and no new large volumes can be created. Any volumes or file systems on a large volume in this situation are also unavailable. If a system with large volumes is rebooted under a version of Solaris prior to Solaris 10, Solaris Volume Manager will not start. All large volumes must be removed before Solaris Volume Manager runs under another version of the Solaris Operating Environment.

オプション

Root privileges are required for any of the following options except `-i`.

The following options are supported:

`-a all` *component*	Add *component* to all hot spare pools. `all` is not case sensitive.
`-a` *hot_spare_pool* [*component*]	Add the *component* to the specified *hot_spare_pool*. *hot_spare_pool* is created if it does not already exist.

`-d all` *component*	Delete *component* from all the hot spare pools. The *component* cannot be deleted if it is in the in-use state.
`-d` *hot_spare_pool* `[`*component*`]`	Delete *hot_spare_pool*, if the *hot_spare_pool* is both empty and not referenced by a metadevice. If *component* is specified, it is deleted from the *hot_spare_pool*. Hot spares in the in-use state cannot be deleted.
`-e` *component*	Enable *component* to be available for use as a hot spare. The *component* can be enabled if it is in the broken state and has been repaired.
`-i [`*hot_spare_pool* `. . .]`	Display the status of the specified *hot_spare_pool* or for all hot spare pools if one is not specified.
`-r all` *component-old component-new*	Replace *component-old* with *component-new* in all hot spare pools which have the component associated. Components cannot be replaced from any hot spare pool if the old hot spare is in the in-use state.
`-r` *hot_spare_pool component-old component-new*	Replace *component-old* with *component-new* in the specified *hot_spare_pool*. Components cannot be replaced from a hot spare pool if the old hot spare is in the in-use state.
`-s` *setname*	Specify the name of the diskset on which `metahs` works. Using the `-s` option causes the command to perform its administrative function within the specified diskset. Without this option, the command performs its function on local hot spare pools.

オペランド

The following operands are supported:

component The logical name for the physical slice (partition) on a disk drive, such as `/dev/dsk/c0t0d0s2`.

hot_spare_pool Names for hot spare pools can be any legal file name that is composed of alphanumeric characters, a dash (“-”), an underscore (“_”), or a period

(“.“). Names must begin with a letter. The words “all” and “none” are reserved and cannot be used.

使用例

例 1 Adding a Hot Spare to a Hot Spare Pool

The following example adds a hot spare `/dev/dsk/c0t0d0s7` to a hot spare pool `mirror1_pool`:

```
# metahs -a mirror1_pool c0t0d0s7
```

When the hot spare is added to the pool, the existing order of the hot spares already in the pool is preserved. The new hot spare is added at the end of the list of hot spares in the hot spare pool specified.

例 2 Adding a Hot Spare to All Currently Defined Pools

This example adds a hot spare to the hot spare pools that are currently defined:

```
# metahs -a all c0t0d0s7
```

The keyword `all` in this example specifies adding the hot spare, `/dev/dsk/c0t0d0s7`, to all the hot spare pools.

例 3 Deleting a Hot Spare

This example deletes a hot spare, `/dev/dsk/c0t0d0s7`, from a hot spare pool, `hsp003`:

```
# metahs -d hsp003 c0t0d0s7
```

When you delete a hot spare, the position of the remaining hot spares in the pool changes to reflect the new order. For instance, if in this example `/dev/dsk/c0t0d0s7` were the second of three hot spares, after deletion the third hot spare would move to the second position.

例 4 Replacing a Hot Spare

This example replaces a hot spare that was previously defined:

```
# metahs -r hsp001 c0t1d0s0 c0t3d0s0
```

In this example, the hot spare `/dev/dsk/c0t1d0s0` is replaced by `/dev/dsk/c0t3d0s0`. The order of the hot spares does not change.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

警告

Do not create large (>1 TB) volumes if you expect to run the Solaris Operating Environment with a 32-bit kernel or if you expect to use a version of the Solaris Operating Environment prior to Solaris 10.

名前

metaimport – imports disk sets into existing Solaris Volume Manager configurations

形式

```
metaimport -s setname [-n] [-f] [-v] [disks]...
metaimport -r [-v] [disks]...
metaimport -V
metaimport -?
```

機能説明

The `metaimport` command allows the importing of disk sets, including replicated disk sets, into an existing Solaris Volume Manager configuration. Replicated disk sets are disk sets created using remote replication software.

The default Solaris Volume Manager configuration specifies a maximum number of disk sets that can be configured. The `metaimport` command fails if importing the disk set would result in exceeding the number of disk sets configured on the system. To increase the number of disk sets allowed on a system, see the 『Solaris ボリュームマネージャの管理』.

Use `metaset(1M)` or `metastat(1M)` to view the configuration of the imported set.

You must run `metaimport` as root.

`metaimport` requires a functional Solaris Volume Manager configuration before it runs.

オプション

The following options are supported:

`-f` Force the import, even if a quorum of replicas from the imported disk set is not available. This option could result in corrupt configurations and should only be used when `metaimport` fails with the "Insufficient quorum detected; exiting" error. If only a partial disk set is available, this option might be necessary to successfully import. Some or all data could be corrupted or unavailable when importing a partial set or a set lacking a replica quorum.

`-n` Does not actually perform the operation, but shows the output or errors that would have resulted from the operation, had it been run.

`-r` Report on the non-configured disk sets found on the system. If no disk device or LUN is specified, `metaimport` reports on all non-configured disk sets attached to the system. When the name of one disk is specified, `metaimport` reports on the disk set (or virtual LUN) containing the specified disk. If two or more disks are specified, metaimport reports on the set (or sets, if they belong to different disk sets) containing the specified disks. If two or more disks are specified, `metaimport` reports on the set (or sets, if they belong to different disk sets) containing the specified disks.

This option can be used in conjunction with the `-v` option to give verbose output on each disk set reported.

`-s` *setname*	Specify the disk set name to use when importing. The imported disk set will be called *setname*, without regard to the name it may have had on a different system.
`-v`	Verbose. Provides detailed information about the `metadb` replica location and status. It also provides detailed information about the disk set configuration and status similar to the "`metastat –c`" output.
`-V`	Version information.
`-?`	Display a help message.

使用例

例 1 Importing a Disk Set

The following example creates a disk set called `blue` and identifies `c1t5d0` as a disk containing a state database replica from the disk set being imported.

```
# metaimport -s blue c1t5d0
```

例 2 Reporting Disk Sets to Import

The following example scans all disks and LUNs attached to the system and configured as part of the system. It scans for disks that could be part of a disk set to be imported. Components that are already part of the Solaris Volume Manager configuration are ignored.

This use of `metaimport` provides suggested forms of the `metaimport` command to use to actually import the disk sets that have been found. You can specify a component on the command line to reduce the scope of the scan and generate results more quickly.

```
# metaimport -r
```

終了ステータス

`0`	Successful completion.
`>0`	An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metareplace(1M)`, `metaset(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `attributes(5)`

『Solaris ボリュームマネージャの管理』

名前

metainit – configure metadevices

形式

```
/usr/sbin/metainit -h

/usr/sbin/metainit [generic options] concat/stripe numstripes width
     component... [-i interlace]

/usr/sbin/metainit [width component... [-i interlace]]
     [-h hot_spare_pool]

/usr/sbin/metainit [generic options] mirror -m submirror
     [read_options] [write_options]
     [pass_num]

/usr/sbin/metainit [generic options] RAID -r component...
     [-i interlace]
     [-h hot_spare_pool] [-k] [-o original_column_count]

/usr/sbin/metainit [generic options] hot_spare_pool
     [hotspare...]

/usr/sbin/metainit [generic options] metadevice-name

/usr/sbin/metainit [generic options] -a

/usr/sbin/metainit [generic options] softpart -p [-e] component
     [-A alignment] size

/usr/sbin/metainit -r
```

機能説明

The `metainit` command configures metadevices and hot spares according to the information specified on the command line. Alternatively, you can run `metainit` so that it uses configuration entries you specify in the `/etc/lvm/md.tab` file (see `md.tab(4)`). All metadevices must be set up by the `metainit` command before they can be used.

Solaris Volume Manager supports storage devices and logical volumes greater than 1 terabyte (TB) when a system runs a 64-bit Solaris kernel. Support for large volumes is automatic. If a device greater than 1 TB is created, Solaris Volume Manager configures it appropriately and without user intervention.

If a system with large volumes is rebooted under a 32–bit Solaris kernel, the large volumes are visible through `metastat` output. Large volumes cannot be accessed, modified or deleted, and no new large volumes can be created. Any volumes or file systems on a large volume in this situation are unavailable. If a system with large volumes is rebooted under a version of Solaris prior to the Solaris 9 4/03 release, Solaris Volume Manager does not start. You must remove all large volumes before Solaris Volume Manager runs under an earlier version of the Solaris Operating System.

If you edit the `/etc/lvm/md.tab` file to configure metadevices, specify one complete configuration entry per line. You then run the `metainit` command with either the `-a` option, to activate all metadevices you entered in the `/etc/lvm/md.tab` file, or with the metadevice name corresponding to a specific configuration entry.

metainit does not maintain the state of the volumes that would have been created when metainit is run with both the -a and -n flags. Any volumes in md.tab that have dependencies on other volumes in md.tab are reported as errors when metainit -a -n is run, although the operations might succeed when metainit -a is run. See md.tab(4).

Solaris Volume Manager never updates the /etc/lvm/md.tab file. Complete configuration information is stored in the metadevice state database, not md.tab. The only way information appears in md.tab is through editing it by hand.

When setting up a disk mirror, the *first* step is to use metainit create a one-on-one concatenation for the named slice. See EXAMPLES.

オプション

The following options are supported:

Generic Options

Root privileges are required for all of the following options except -h.

The following generic options are supported:

-f
: Forces the metainit command to continue even if one of the slices contains a mounted file system or is being used as swap, or if the stripe being created is smaller in size than the underlying soft partition. This option is required when configuring mirrors on swap.

-h
: Displays usage message.

-n
: Checks the syntax of your command line or md.tab entry without actually setting up the metadevice. If used with -a, all devices are checked but not initialized.

-r
: Only used in a shell script at boot time. Sets up all metadevices that were configured before the system crashed or was shut down. The information about previously configured metadevices is stored in the metadevice state database (see metadb(1M)).

-s *setname*
: Specifies the name of the diskset on which metainit works. Without the -s option, the metainit command operates on your local metadevices and/or hotspares.

Concat/Stripe Options

The following concat/stripe options are supported:

concat/stripe
: Specifies the metadevice name of the concatenation, stripe, or concatenation of stripes being defined.

numstripes
: Specifies the number of individual stripes in the metadevice. For a simple stripe, *numstripes* is always 1. For a concatenation, *numstripes* is equal to the number of slices. For a concatenation of stripes, *numstripes* varies according to the number of stripes.

width
Specifies the number of slices that make up a stripe. When *width* is greater than 1, the slices are striped.

component
The logical name for the physical slice (partition) on a disk drive, such as `/dev/dsk/c0t0d0s0`. For RAID level 5 metadevices, a minimum of three slices is necessary to enable striping of the parity information across slices.

`-i` *interlace*
Specifies the interlace size. This value tells Solaris Volume Manager how much data to place on a slice of a striped or RAID level 5 metadevice before moving on to the next slice. *interlace* is a specified value, followed by either ‘`k`' for kilobytes, ‘`m`' for megabytes, or ‘`b`' for blocks. The characters can be either uppercase or lowercase. The *interlace* specified cannot be less than 16 blocks, or greater than 100 megabytes. If *interlace* is not specified, it defaults to 512 kilobytes.

`-h` *hot_spare_pool*
Specifies the *hot_spare_pool* to be associated with the metadevice. If you use the command line, the hot spare pool must have been previously created by the `metainit` command before it can be associated with a metadevice. Use `/-h hsp`*nnn* when the concat/stripe being created is to be used as a submirror.

Names for hot spare pools can be any legal file name that is composed of alphanumeric characters, a dash (“-”), an underscore (“_”), or a period (“.”). Names must begin with a letter. The words “all” and “none“ are reserved and cannot be used.

Mirror Options

The following mirror options are supported:

mirror `-m` *submirror*
Specifies the metadevice name of the mirror. The `-m` indicates that the configuration is a mirror. *submirror* is a metadevice (stripe or concatentation) that makes up the initial one-way mirror. Solaris Volume Manager supports a maximum of four-way mirroring. When defining mirrors, first create the mirror with the `metainit` command as a one-way mirror. Then attach subsequent submirrors using the `metattach` command. This method ensures that Solaris Volume Manager properly syncs the mirrors. (The second and any subsequent submirrors are first created using the `metainit` command.)

read_options
The following read options for mirrors are supported:

`-g`
Enables the geometric read option, which results in faster performance on sequential reads.

`-r`
Directs all reads to the first submirror. This should only be used when the devices comprising the first submirror are substantially faster than those of the second mirror. This flag cannot be used with the `-g` flag.

If neither the -g nor -r flags are specified, reads are made in a round-robin order from all submirrors in the mirror. This enables load balancing across the submirrors.

write_options
The following write options for mirrors are supported:

-S
Performs serial writes to mirrors. The first submirror write completes before the second is started. This can be useful if hardware is susceptible to partial sector failures. If -S is not specified, writes are replicated and dispatched to all mirrors simultaneously.

pass_num
A number in the range 0-9 at the end of an entry defining a mirror that determines the order in which that mirror is resynced during a reboot. The default is 1. Smaller pass numbers are resynced first. Equal pass numbers are run concurrently. If 0 is used, the resync is skipped. 0 should be used only for mirrors mounted as read-only, or as swap.

RAID Level 5 Options

The following RAID level 5 options are available:

RAID -r
Specifies the name of the RAID level 5 metadevice. The -r specifies that the configuration is RAID level 5.

-k
For RAID level 5 metadevices, informs the driver that it is not to initialize (zero the disk blocks) due to existing data. Only use this option to recreate a previously created RAID level 5 device.

Use the -k option with extreme caution. This option sets the disk blocks to the OK state. If any errors exist on disk blocks within the metadevice, Solaris Volume Manager might begin fabricating data. Instead of using the -k option, you might want to initialize the device and restore data from tape.

-o *original_column_count*
For RAID level 5 metadevices, used with the -k option to define the number of original slices in the event the originally defined metadevice was grown. This is necessary since the parity segments are not striped across concatenated devices.

Use the -o option with extreme caution. This option sets the disk blocks to the OK state. If any errors exist on disk blocks within the metadevice, Solaris Volume Manager might begin fabricating data. Instead of using the -o option, you might want to initialize the device and restore data from tape.

Soft Partition Options

The following soft partition options are supported:

softpart -p [-e] *component* [-A *alignment*] *size*
The *softpart* argument specifies the name of the soft partition. The -p specifies that the configuration is a soft partition.

The `-e` specifies that the entire disk specified by *component* as `c*t*d*` should be repartitioned and reserved for soft partitions. The specified component is repartitioned such that slice 7 reserves space for system (state database replica) usage and slice 0 contains all remaining space on the disk. Slice 7 is a minimum of 4MB, but can be larger, depending on the disk geometry. The newly created soft partition is placed on slice 0 of the device.

The *component* argument specifies the disk (`c*t*d*`), slice (`c*t*d*s*`), or meta device (`d*`) from which to create the soft partition. The *size* argument determines the space to use for the soft partition and can be specified in `K` or `k` for kilobytes, `M` or `m` for megabytes, `G` or `g` for gigabytes, `T` or `t` for terabyte (one terabyte is the maximum size), and `B` or `b` for blocks (sectors). All values represent powers of 2, and upper and lower case options are equivalent. Only integer values are permitted.

The `-A` alignment option sets the value of the soft partition extent alignment. This option used when it is important specify a starting offset for the soft partition. It preserves the data alignment between the metadevice address space and the address space of the underlying physical device. For example, a hardware device that does checksumming should not have its I/O requests divided by Solaris Volume Manager. In this case, use a value from the hardware configuration as the value for the alignment. When you use this option in conjunction with a software I/O load, the alignment value corresponds to the I/O load of the application. This prevents I/O from being divided unnecessarily and affecting performance.

The literal `all`, used instead of specifying size, specifies that the soft partition should occupy all available space on the device.

Hot Spare Pool Options

The following hot spare pool options are supported:

hot_spare_pool [*hotspare...*]
: When used as arguments to the `metainit` command, *hot_spare_pool* defines the name for a hot spare pool, and *hotspare...* is the logical name for the physical slice(s) for availability in that pool. Names for hot spare pools can be any legal file name that is composed of alphanumeric characters, a dash ("-"), an underscore ("_"), or a period (".”). Names must begin with a letter. The words “all“ and “none“ are reserved and cannot be used.

md.tab File Options

The following `md.tab` file options are supported:

metadevice-name
: When the `metainit` command is run with a *metadevice-name* as its only argument, it searches the `/etc/lvm/md.tab` file to find that name and its corresponding entry. The order in which entries appear in the `md.tab` file is unimportant. For example, consider the following `md.tab` entry:

```
d0 2 1 c1t0d0s0 1 c2t1d0s0
```

When you run the command `metainit d0`, it configures metadevice `d0` based on the configuration information found in the `md.tab` file.

`-a`

Activates all metadevices defined in the `md.tab` file.

`metainit` does not maintain the state of the volumes that would have been created when `metainit` is run with both the `-a` and `-n` flags. If a device `d0` is created in the first line of the `md.tab` file, and a later line in `md.tab` assumes the existence of `d0`, the later line fails when `metainit -an` runs (even if it would succeed with `metainit -a`).

使用例

例 1 Creating a One-on-One Concatenation

The following command creates a one-on-one concatenation for the root slice. This is the first step you take when setting up a mirror for any other slice that cannot be unmounted. The `-f` option is required to create a volume with an existing file system.

```
# metainit -f dl 1 1 c0t0d0s0
```

The preceding command makes `d1` a one-on-one concatenation, using the named slice. You can then enter:

```
# metainit d0 -m d1
```

...to make a one-way mirror of the named slice.

例 2 Concatenation

All drives in the following examples have the same size of 525 Mbytes.

This example shows a metadevice, `/dev/md/dsk/d7`, consisting of a concatenation of four slices.

```
# metainit d7 4 1 c0t1d0s0 1 c0t2d0s0 1 c0t3d0s0 1 /dev/dsk/c0t4d0s0
```

The number 4 indicates there are four individual stripes in the concatenation. Each stripe is made of one slice, hence the number 1 appears in front of each slice. The first disk sector in all of these devices contains a disk label. To preserve the labels on devices `/dev/dsk/c0t2d0s0`, `/dev/dsk/c0t3d0s0`, and `/dev/dsk/c0t4d0s0`, the metadisk driver must skip at least the first sector of those disks when mapping accesses across the concatenation boundaries. Because skipping only the first sector would create an irregular disk geometry, the entire first cylinder of these disks is skipped. This allows higher level file system software to optimize block allocations correctly.

例 3 Stripe

This example shows a metadevice, `/dev/md/dsk/d15`, consisting of two slices.

```
# metainit d15 1 2 c0t1d0s0 c0t2d0s0 -i 32k
```

The number 1 indicates that one stripe is being created. Because the stripe is made of two slices, the number 2 follows next. The optional `-i` followed by `32k` specifies the interlace size as 32 Kbytes. If the interlace size were not specified, the stripe would use the default value of 16

例 3 Stripe (続き)

Kbytes.

例 4 Concatentation of Stripes

This example shows a metadevice, /dev/md/dsk/d75, consisting of a concatenation of two stripes of three disks.

```
# metainit d75 2 3 c0t1d0s0 c0t2d0s0 \
      c0t3d0s0 -i 16k \
      3 c1t1d0s0 c1t2d0s0 c1t3d0s0 -i 32k
```

On the first line, the -i followed by 16k specifies that the stripe interlace size is 16 Kbytes. The second set specifies the stripe interlace size as 32 Kbytes. If the second set did not specify 32 Kbytes, the set would use the default interlace value of 16 Kbytes. The blocks of each set of three disks are interlaced across three disks.

例 5 Mirroring

This example shows a two-way mirror, /dev/md/dsk/d50, consisting of two submirrors. This mirror does not contain any existing data.

```
# metainit d51 1 1 c0t1d0s0
# metainit d52 1 1 c0t2d0s0
# metainit d50 -m d51
# metattach d50 d52
```

In this example, two submirrors, d51 and d52, are created with the metainit command. These two submirrors are simple concatenations. Next, a one-way mirror, d50, is created using the -m option with d51. The second submirror is attached later using the metattach command. When creating a mirror, any combination of stripes and concatenations can be used. The default read and write options in this example are a round-robin read algorithm and parallel writes to all submirrors.

例 6 Creating a metadevice in a diskset

This example shows a metadevice, /dev/md/dsk/d75, consisting of a concatenation of two stripes within a diskset called set1.

```
# metainit -s set1 d75 2 3 c2t1d0s0 c2t2d0s0 \
      c2t3d0s0 -i 32k
# metainit -s set1 d51 1 1 c2t1d0s0
# metainit -s set1 d52 1 1 c3t1d0s0
# metainit -s set1 d50 -m d51
# metattach -s set1 d50 d52
```

In this example, a diskset is created using the metaset command. Metadevices are then created within the diskset using the metainit command. The two submirrors, d51 and d52, are simple concatenations. Next, a one-way mirror, d50, is created using the -m option with

例 6 Creating a metadevice in a diskset (続き)

d51. The second submirror is attached later using the metattach command. When creating a mirror, any combination of stripes and concatenations can be used. The default read and write options in this example are a round-robin read algorithm and parallel writes to all submirrors.

例 7 RAID Level 5

This example shows a RAID level 5 device, d80, consisting of three slices:

```
# metainit d80 -r c1t0d0s0 c1t1d0s0 c1t3d0s0 -i 20k
```

In this example, a RAID level 5 metadevice is defined using the -r option with an interlace size of 20 Kbytes. The data and parity segments are striped across the slices, c1t0d0s0, c1t2d0s0, and c1t3d0s0.

例 8 Soft Partition

The following example shows a soft partition device, d1, built on metadevice d100 and 100 Mbytes (indicated by 100M) in size:

```
# metainit d1 -p d100 100M
```

The preceding command creates a 100 Mbyte soft partition on the d100 metadevice. This metadevice could be a RAID level 5, stripe, concatenation, or mirror.

例 9 Soft Partition on Full Disk

The following example shows a soft partition device, d1, built on disk c3t4d0:

```
# metainit d1 -p -e c3t4d0 9G
```

In this example, the disk is repartitioned and a soft partition is defined to occupy all 9 Gbytes of disk c3t4d0s0.

例 10 Soft Partition Taking All Available Space

The following example shows a soft partition device, d1, built on disk c3t4d0:

```
# metainit d1 -p -e c3t4d0 all
```

In this example, the disk is repartitioned and a soft partition is defined to occupy all available disk space on slice c3t4d0s0.

例 11 Hot Spare

This example shows a two-way mirror, /dev/md/dsk/d10, and a hot spare pool with three hot spare components. The mirror does not contain any existing data.

```
# metainit hsp001 c2t2d0s0 c3t2d0s0 c1t2d0s0
# metainit d41 1 1 c1t0d0s0 -h hsp001
# metainit d42 1 1 c3t0d0s0 -h hsp001
```

例 11 Hot Spare (続き)

```
# metainit d40 -m d41
# metattach d40 d42
```

In this example, a hot spare pool, hsp001, is created with three slices from three different disks used as hot spares. Next, two submirrors are created, d41 and d42. These are simple concatenations. The metainit command uses the -h option to associate the hot spare pool hsp001 with each submirror. A one-way mirror is then defined using the -m option. The second submirror is attached using the metattach command.

例 12 Setting the Value of the Soft Partition Extent Alignment

This example shows how to set the alignment of the soft partition to 1 megabyte.

```
# metainit -s red d13 -p c1t3d0s4 -A 1m 4m
```

In this example the soft partition, d13, is created with an extent alignment of 1 megabyte. The metainit command uses the -A option with an alignment of 1m to define the soft partition extent alignment.

ファイル

/etc/lvm/md.tab
: Contains list of metadevice and hot spare configurations for batch-like creation.

警告

This section contains information on different types of warnings.

Devices and Volumes Greater Than 1 TB

Do not create large (>1 TB) volumes if you expect to run the Solaris Operating Environment with a 32–bit kernel or if you expect to use a version of the Solaris Operating Environment prior to Solaris 10.

Multi-Way Mirror

Do not use the metainit command to create a multi-way mirror. Rather, create a one-way mirror with metainit then attach additional submirrors with metattach. When the metattach command is not used, no resync operations occur and data could become corrupted.

If you use metainit to create a mirror with multiple submirrors, the following message is displayed:

```
WARNING: This form of metainit is not recommended.
The submirrors may not have the same data.
Please see ERRORS in metainit(1M) for additional information.
```

Truncation of Soft Partitions

When creating stripes on top of soft partitions it is possible for the size of the new stripe to be less than the size of the underlying soft partition. If this occurs, metainit fails with an error indicating the actions required to overcome the failure.

If you use the -f option to override this behavior, the following message is displayed:

```
WARNING: This form of metainit is not recommended.
The stripe is truncating the size of the underlying device.
Please see ERRORS in metainit(1M) for additional information.
```

Write-On-Write Problem

When mirroring data in Solaris Volume Manager, transfers from memory to the disks do not all occur at exactly the same time for all sides of the mirror. If the contents of buffers are changed while the data is in-flight to the disk (called write-on-write), then different data can end up being stored on each side of a mirror.

This problem can be addressed by making a private copy of the data for mirror writes, however, doing this copy is expensive. Another approach is to detect when memory has been modified across a write by looking at the dirty-bit associated with the memory page. Solaris Volume Manager uses this dirty-bit technique when it can. Unfortunately, this technique does not work for raw I/O or direct I/O. By default, Solaris Volume Manager is tuned for performance with the liability that mirrored data might be out of sync if an application does a "write-on-write" to buffers associated with raw I/O or direct I/O. Without mirroring, you were not guaranteed what data would actually end up on media, but multiple reads would return the same data. With mirroring, multiple reads can return different data. The following line can be added to `/etc/system` to cause a stable copy of the buffers to be used for all raw I/O and direct I/O write operations.

```
set md_mirror:md_mirror_wow_flg=0x20
```

Setting this flag degrades performance.

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metarename(1M)`, `metareplace(1M)`, `metaset(1M)`, `metassist(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `md(7D)`

Limitations

Recursive mirroring is not allowed; that is, a mirror cannot appear in the definition of another mirror.

Recursive logging is not allowed; that is, a trans metadevice cannot appear in the definition of another metadevice.

Stripes, concatenations, and RAID level 5 metadevices must consist of slices only.

Mirroring of RAID level 5 metadevices is not allowed.

Soft partitions can be built on raw devices, or on stripes, RAID level 5, or mirrors.

RAID level 5 or stripe metadevices can be built directly on soft partitions.

注意事項

Trans metadevices have been replaced by UFS logging. Existing trans devices are *not* logging--they pass data directly through to the underlying device. See `mount_ufs(1M)` for more information about UFS logging.

名前

metaoffline, metaonline – place submirrors offline and online

形式

```
/usr/sbin/metaoffline -h
/usr/sbin/metaoffline [-s setname] [-f] mirror submirror
/usr/sbin/metaonline -h
/usr/sbin/metaonline [-s setname] mirror submirror
```

機能説明

The `metaoffline` command prevents Solaris Volume Manager from reading and writing to the submirror that has been taken offline. While the submirror is offline, all writes to the mirror will be kept track of (by region) and will be written when the submirror is brought back online. The `metaoffline` command can also be used to perform online backups: one submirror is taken offline and backed up while the mirror remains accessible. (However, if this is a two-way mirror, data redundancy is lost while one submirror is offline.) The `metaoffline` command differs from the `metadetach` command because it does not sever the logical association between the submirror and the mirror. To completely remove a submirror from a mirror, use the `metadetach` command.

A submirror that has been taken offline will only remain offline until the `metaonline` command is invoked or the system is rebooted.

When the `metaonline` command is used, reading from and writing to the submirror resumes. A resync is automatically invoked to resync the regions written while the submirror was offline. Writes are directed to the submirror during resync. Reads, however, will come from a different submirror. Once the resync operation completes, reads and writes are performed on that submirror. The `metaonline` command is only effective on a submirror of a mirror that has been taken offline.

The `metaoffline` and `metaonline` commands can not be used on RAID 1 volumes in application-based recovery (ABR) mode.

A submirror that has been taken offline with the `metaoffline` command can only be mounted as read-only.

オプション

Root privileges are required for all of the following options except `-h`.

`-f`	Forces offlining of submirrors that have slices requiring maintenance.
`-h`	Displays usage message.
`-s` *setname*	Specifies the name of the diskset on which `metaoffline` and `metaonline` will work. Using the `-s` option will cause the command to perform its administrative function within the specified diskset. Without this option, the command will perform its function on local metadevices.
mirror	Specifies the metadevice name of the mirror from which the submirror will be either taken offline or put online.

submirror Specifies the metadevice name of the submirror to be either taken offline or put online.

使用例

例 1 Taking a Submirror Offline

This example takes one submirror, `mymirror_sub1`, offline from mirror `mymirror`.

```
# metaoffline mymirror mymirror_sub1
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metarename(1M)`, `metareplace(1M)`, `metaset(1M)`, `metassist(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `md(7D)`

『Solaris ボリュームマネージャの管理』

注意事項

The `metaonline` and `metaoffline` commands are not applicable to mirrors in application-based recovery (ABR) mode.

名前

metaparam – modify parameters of metadevices

形式

```
/usr/sbin/metaparam -h
```

```
/usr/sbin/metaparam [-s setname]
    [concat/stripe or RAID5 options] concat/stripe RAID
```

```
/usr/sbin/metaparam [-s setname] [mirror options] mirror
```

機能説明

The `metaparam` command is used to display or modify current parameters of metadevices.

If just the metadevice is specified as an argument to the `metaparam` command, the current settings are displayed.

The `metaparam` command enables most metadevice (volume) parameters to be changed. Only the interlace value cannot be changed by `metaparam`, because it is established when the metadevice is created and cannot be changed thereafter.

オプション

Root privileges are required for all of the options.

The following options are supported:

`-h`
: Displays usage message.

`-s` *setname*
: Specify the name of the diskset on which `metaparam` works. Using the `-s` option causes the command to perform its administrative function within the specified diskset. Without this option, the command performs its function on local metadevices.

Concat/STRIPE Or Raid5 Options

–h*hot_spare_pool* | `none`
: Specifies the hot spare pool to be used by a metadevice. If `none` is specified, the metadevice is disassociated with the hot spare pool assigned to it. If the metadevice is currently using a hot spare, then `metaparam` cannot replace the hot spare pool.

concat/stripe | *RAID*
: Specifies the metadevice name of the concatenation, stripe, or concatenation of stripes, or of the RAID5 metadevice.

Mirror Options

–`r roundrobin | geometric | first`
: Modifies the read option for a mirror. The `-r` option must be followed by either `roundrobin`, `geometric`, or `first`. `roundrobin`, which is the default action under the `metainit` command, specifies reading the disks in a round-robin (load balancing) method. `geometric` allows for faster performance on sequential reads. `first` specifies reading only from the first submirror.

–`w parallel | serial`
: Modifies the write option for a mirror. The `-w` option must be followed by either `parallel` or `serial`. `parallel`, the default action under the `metainit` command, specifies that all writes are parallel. `serial` specifies that all writes are serial.

-p *pass_number*
A number from 0-to-9 that specifies the order in which a mirror is resynced during reboot. The default is 1. Smaller pass numbers are resynced first. Equal pass numbers are run concurrently. If 0 is used, the mirror resync is skipped. 0 should only be used for mirrors mounted as read-only, or as swap.

mirror
Specifies the metadevice name of the mirror.

使用例

例 1 Associating Hot Spare Pool with RAID5 Metadevice

This example associates a hot spare pool, user_pool, with a RAID5 metadevice, user_raid.

```
# metaparam -h user_pool user_raid
```

例 2 Changing Read Option to Geometric

This example changes the read option on a mirror d50 from the default of roundrobin to geometric.

```
# metaparam -r geometric d50
```

終了ステータス

The following exit values are returned:

0
Successful completion.

>0
An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

名前

metarecover – recover soft partition information

形式

```
/usr/sbin/metarecover [-n] [-v] [-s setname] component -p
/usr/sbin/metarecover [-n] [-v] [-s setname] component -p {-d}
/usr/sbin/metarecover [-n] [-v] [-s setname] component -p {-m}
```

機能説明

The `metarecover` command scans a specified component to look for soft partition configuration information and to regenerate the configuration.

オプション

The following options are supported:

`-d` Recover soft partitions in the metadevice state database from the extent headers on the device. Options `-d` and `-m` are mutually exclusive.

`-m` Regenerate the extent headers and reapplies them to the underlying device based on the soft partitions listed in the metadevice state database. Options `-d` and `-m` are mutually exclusive.

`-n` Do not actually perform the operation. Show the output or errors that would have resulted from the operation, had it been run.

`-p` Regenerate soft partitions based on the metadevice state database or extent headers on the underlying device. If neither `-d` nor `-m` are specified, this option compares the soft partition information in the metadevice state database to the extent headers.

`-s` *setname* Specify the name of the diskset on which `metarecover` works. Using the `s` option causes the command to perform its function within the specified diskset. Without the `-s` option, the `metarecover` command operates on the metadevices and/or hot spare pools in the local diskset.

This option is required to recover former `sps` from a diskset component or raw-device. *setname* must be identical to the former *setname* in which the `sps` were created. The set numbers, however, seem irrelevant.

`-v` Verbose mode, displaying the changes being made.

オペランド

The following operand is supported:

component Specifies the `c*t*d*s*` number of the disk or slice containing the partitions, or the device name (for example, `d10`) of the metadevice containing the partitions.

component can be a slice name, component name, `/dev/dsk path`, or `/dev/rdsk path`.

使用例

例 1 Updating Metadevice State Database Based on Disk Extent Headers

A disk containing soft partitions is moved from one system to another. The system administrator would like to use the existing soft partitions. `metarecover` updates the metadevice state database based on the extent headers on the disk.

```
# metarecover -v c0t3d0s2 -p -d
```

例 2 Updating Metadevice State Database Based on Incomplete Soft Partition Creation

A system crashes in the middle of creating a new soft partition. The soft partition is in the creating state and the driver does not let that device be opened. `metarecover` rewrites the extent headers for the partially created soft partition and mark it as `Okay`.

```
# metarecover -v c0t3d0s2 -p -m
```

例 3 Updating Extent Headers Based on Metadevice State Database

Someone accidentally overwrote a portion of a disk leaving extent headers destroyed. `metarecover` rewrites the extent headers to ensure a valid soft partition configuration, though user data is not recovered.

```
# metarecover -v d5 -m
```

The following example implements the same command using a descriptive name.

```
# metarecover -v myvolume -m
```

例 4 Validating Soft Partition Configuration

To validate the existing soft partition configuration, use `metarecover` with only the `-p` flag.

```
# metarecover c0t3d0s2 -p
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarename(1M)`, `metareplace(1M)`, `metaset(1M)`, `metassist(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `md(7D)`

『Solaris ボリュームマネージャの管理』

名前

metarename – rename metadevice or switch layered metadevice names

形式

```
/usr/sbin/metarename [-s setname] metadevice1 metadevice2
/usr/sbin/metarename [-s setname] [-f] -x metadevice1 metadevice2
/usr/sbin/metarename -h
```

機能説明

There are two ways to use `metarename`, one with and one without the `-x` option. The first method (without `-x`) renames an existing metadevice to a new name. This makes managing the metadevice namespace easier. The metadevice being renamed cannot be mounted or open, nor can the new name already exist. For example, to rename a metadevice that contains a mounted file system, you would first need to unmount the file system.

With the second way to use `metarename`, using the `-x` option, `metarename` switches (exchanges) the names of an existing layered metadevice and one of its subdevices. In Solaris Volume Manager terms, a layered metadevice can be either a mirror or a trans metadevice. The `-x` option enables you to switch the metadevice names of a mirror and one of its submirrors, or a trans metadevice and its master device.

`metarename -x` makes it easier to mirror or unmirror an existing stripe or concatenation, and to remove a trans device.

When used to mirror an existing stripe or concatenatation, you must stop access to the device. For example, if the device contains a mounted file system, you must first unmount the file system before doing the rename.

You can also use the `metarename -x` command to untrans a trans metadevice from an existing device. This applies only to the master device. You cannot remove a logging device with `metarename`. Before you can rename a trans device, you must detach the logging device. Then you must stop access to the trans metadevice itself.

You cannot rename or switch metadevices that are in an error state or that have subcomponents in an error state, or metadevices actively using a hot spare replacement.

You can only switch metadevices that have a direct child/parent relationship. You could not, for example, directly exchange a stripe in a mirror that is a master device with the trans metadevice.

You must use the `-f` flag when switching members of a trans metadevice.

Only metadevices can be switched, not slices.

オプション

The following options are supported:

`-f`	Force the switching of trans metadevice members.
`-h`	Display a help message.

`-s` *setname*	Specifies the name of the diskset on which `metarename` will work. Using the `-s` option will cause the command to perform its administrative function within the specified diskset. Without this option, the command will perform its function on the local metadevices.
`-x`	Exchange the metadevice names *metadevice1* and *metadevice2*.
metadevice1	Specifies the metadevice to be renamed or switched.
metadevice2	Specifies the target metadevice name for the rename or switch operation.

使用例

例 1 Renaming a Metadevice

This example renames a metadevice named `d10` to `account_records`. Note that `account_records` must not exist for the rename to succeed.

```
# metarename d10 account_records
```

例 2 Creating a Two-Way Mirror

This example creates a two-way mirror from an existing stripe named `d1` with a mounted file system, `/home2`.

```
# metainit d2 1 1 c13d0s1
# metainit -f d20 -m d1
# umount /home2
# metarename -x d20 d1
# metattach d1 d2
# mount /home2
```

First, a second concatenation `d2`, is created. (`d1` already exists.) The `metainit` command creates a one-way mirror, `d20`, from `d1`. Next, you umount the file system and switch `d1` for `d20`, making `d1` the top-level device (mirror). You attach the second submirror, `d2`, to create a two-way mirror. Lastly, you remount the file system.

例 3 Mounting a Mirrored File System on Stripe

This example takes an existing mirror named `d1` with a mounted file system, and ends up with the file system mounted on a stripe `d1`.

```
# umount /fs2
# metarename -x d1 d20
# metadetach d20 d1
# metaclear -r d20
# mount /fs2
```

First, you unmount the file system, then switch the mirror `d1` and its submirror `d20`. This makes the mirror into `d20`. Next, you detach `d1` from `d20`, then delete the mirror `d20` and its other submirror. You then remount the file system.

例 4 Deleting a Trans Metadevice

This example deletes a trans metadevice named d10 while its mount point is /myhome. The master device, which is a stripe, is named d2. The logging device, also a stripe, is named d5.

```
# umount /myhome
# metadetach d10
# metarename -f -x d10 d2
# metaclear d2
# metaclear d5
# fsck /dev/md/dsk/d10
# mount /myhome
```

You umount the file system first, then detach the trans metadevice's logging device. The trans metadevice is switched with the master device, making the trans metadevice d2 and the underlying stripe d10. You clear the trans metadevice d2 and the logging device d5. d10 must be fsck'd, and then the file system is remounted.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

Limitations

Renaming and exchanging metadevice names can only be used for metadevices. A physical slice cannot be renamed to a metadevice, nor can a metadevice be exchanged with a physical slice name.

Metadevice names are strings of the pattern *d<xyz>* where *xyz* is a value between 0 and 8192. You cannot use logical names for metadevices.

注意事項

Trans metadevices have been replaced by UFS logging. Existing trans devices are *not* logging--they pass data directly through to the underlying device. See mount_ufs(1M) for more information about UFS logging.

名前

metareplace – enable or replace components of submirrors or RAID5 metadevices

形式

```
/usr/sbin/metareplace -h
/usr/sbin/metareplace [-s setname] -e mirror component
/usr/sbin/metareplace [-s setname] mirror component-old component-new
/usr/sbin/metareplace [-s setname] -e RAID component
/usr/sbin/metareplace [-s setname] [-f] RAID component-old component-new
```

機能説明

The `metareplace` command is used to enable or replace components (slices) within a submirror or a RAID5 metadevice.

When you replace a component, the `metareplace` command automatically starts resyncing the new component with the rest of the metadevice. When the resync completes, the replaced component becomes readable and writable. If the failed component has been hot spare replaced, the hot spare is placed in the available state and made available for other hot spare replacements.

Note that the new component must be large enough to replace the old component.

A component may be in one of several states. The `Last Erred` and the `Maintenance` states require action. Always replace components in the `Maintenance` state first, followed by a resync and validation of data. After components requiring maintenance are fixed, validated, and resynced, components in the `Last Erred` state should be replaced. To avoid data loss, it is always best to back up all data before replacing `Last Erred` devices.

オプション

Root privileges are required for all of the following options except `-h`.

`-e` Transitions the state of *component* to the available state and resyncs the failed component. If the failed component has been hot spare replaced, the hot spare is placed in the available state and made available for other hot spare replacements. This command is useful when a component fails due to human error (for example, accidentally turning off a disk), or because the component was physically replaced. In this case, the replacement component must be partitioned to match the disk being replaced before running the `metareplace` command.

`-f` Forces the replacement of an errored component of a metadevice in which multiple components are in error. The component determined by the `metastat` display to be in the “Maintenance" state must be replaced first. This option may cause data to be fabricated since multiple components are in error.

`-h` Display help message.

`-s` *setname* Specifies the name of the diskset on which `metareplace` will work. Using the `-s` option will cause the command to perform its administrative

function within the specified diskset. Without this option, the command will perform its function on local metadevices.

mirror The metadevice name of the mirror.

component The logical name for the physical slice (partition) on a disk drive, such as `/dev/dsk/c0t0d0s2`.

component-old The physical slice that is being replaced.

component-new The physical slice that is replacing *component-old*.

RAID The metadevice name of the RAID5 device.

使用例

例 1 Recovering from Error Condition in RAID5 Metadevice

This example shows how to recover when a single component in a RAID5 metadevice is errored.

```
# metareplace d10 c3t0d0s2 c5t0d0s2
```

In this example, a RAID5 metadevice `d10` has an errored component, `c3t0d0s2`, replaced by a new component, `c5t0d0s2`.

例 2 Use of -e After Physical Disk Replacement

This example shows the use of the `-e` option after a physical disk in a submirror (a submirror of mirror `mymirror1`, in this case) has been replaced.

```
# metareplace -e mymirror1 c1t4d0s2
```

Note: The replacement disk must be partitioned to match the disk it is replacing before running the `metareplace` command.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`,

metarename(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

名前

metaset – configure disk sets

形式

```
/usr/sbin/metaset -s setname [-M-a -h hostname]

/usr/sbin/metaset -s setname -A {enable | disable}

/usr/sbin/metaset -s setname [-A {enable | disable}] -a -h hostname...

/usr/sbin/metaset -s setname -a [-l length] [-L] drivename...

/usr/sbin/metaset -s setname -C {take | release | purge}

/usr/sbin/metaset -s setname -d [-f] -h hostname...

/usr/sbin/metaset -s setname -d [-f] drivename...

/usr/sbin/metaset -s setname -j

/usr/sbin/metaset -s setname -r

/usr/sbin/metaset -s setname -w

/usr/sbin/metaset -s setname -t [-f] [-u tagnumber] [y]

/usr/sbin/metaset -s setname -b

/usr/sbin/metaset -s setname -P

/usr/sbin/metaset -s setname -q

/usr/sbin/metaset -s setname -o [-h hostname]

/usr/sbin/metaset [-s setname]

/usr/sbin/metaset [-s setname] -a | -d
     [ [m] mediator_host_list]
```

機能説明

The `metaset` command administers sets of disks in named disk sets. Named disk sets include any disk set that is not in the local set. While disk sets enable a high-availability configuration, Solaris Volume Manager itself does not actually provide a high-availability environment.

A single-owner disk set configuration manages storage on a SAN or fabric-attached storage, or provides namespace control and state database replica management for a specified set of disks.

In a shared disk set configuration, multiple hosts are physically connected to the same set of disks. When one host fails, another host has exclusive access to the disks. Each host can control a shared disk set, but only one host can control it at a time.

When you add a new disk to any disk set, Solaris Volume Manager checks the disk format. If necessary, it repartitions the disk to ensure that the disk has an appropriately configured reserved slice 7 (or slice 6 on an EFI labelled device) with adequate space for a state database replica. The precise size of slice 7 (or slice 6 on an EFI labelled device) depends on the disk geometry. For tradtional disk sets, the slice is no less than 4 Mbytes, and probably closer to 6 Mbytes, depending on where the cylinder boundaries lie. For multi-owner disk sets, the slice is

a minimum of 256 Mbytes. The minimal size for slice 7 might change in the future. This change is based on a variety of factors, including the size of the state database replica and information to be stored in the state database replica.

For use in disk sets, disks must have a dedicated slice (six or seven) that meets specific criteria:

- The slice must start at sector 0
- The slice must include enough space for disk label
- The state database replicas cannot be mounted
- The slice does not overlap with any other slices, including slice 2

If the existing partition table does not meet these criteria, or if the -L flag is specified, Solaris Volume Manager repartitions the disk. A small portion of each drive is reserved in slice 7 (or slice 6 on an EFI labelled device) for use by Solaris Volume Manager. The remainder of the space on each drive is placed into slice 0. Any existing data on the disks is lost by repartitioning.

After you add a drive to a disk set, it can be repartitioned as necessary, with the exception that slice 7 (or slice 6 on an EFI labelled device) is not altered in any way.

After a disk set is created and metadevices are set up within the set, the metadevice name is in the following form:

`/dev/md/`*setname*`/{dsk,rdsk}/d`*number*

where *setname* is the name of the disk set, and *number* is the number of the metadevice (`0-127`).

If you have disk sets that you upgraded from Solstice DiskSuite software, the default state database replica size on those sets is 1034 blocks, not the 8192 block size from Solaris Volume Manager. Also, slice 7 on the disks that were added under Solstice DiskSuite are correspondingly smaller than slice 7 on disks that were added under Solaris Volume Manager.

If disks you add to a disk set have acceptable slice 7s (that start at cylinder 0 and that have sufficient space for the state database replica), they are not reformatted.

Hot spare pools within local disk sets use standard Solaris Volume Manager naming conventions. Hot spare pools with shared disk sets use the following convention:

setname/*hot_spare_pool*

where *setname* is the name of the disk set, and *hot_spare_pool* is the name of the hot spare pool associated with the disk set.

Multi–node Environment

To create and work with a disk set in a multi—node environment, `root` must be a member of Group 14 on all hosts, or the `/.rhosts` file must contain an entry for all other host names. This is not required in a SunCluster 3.x enviroment.

Tagged data

Tagged data occurs when there are different versions of a disk set's replicas. This tagged data consists of the set owner's nodename, the hardware serial number of the owner, and the time it was written out to the available replicas. The system administer can use this information to determine which replica contains the correct data.

When a disk set is configured with an even number of storage enclosures and has replicas balanced across them evenly, it is possible that up to half of the replicas can be lost (for example, through a power failure of half of the storage enclosures). After the enclosure that went down is rebooted, half of the replicas are not recognized by SVM. When the set is retaken, the `metaset` command returns an error of "stale databases", and all of the metadevices are in a read-only state.

Some of the replicas that are not recognized need to be deleted. The action of deleting the replicas also causes updates to the replicas that are not being deleted. In a dual hosted disk set environment, the second node can access the deleted replicas instead of the existing replicas when it takes the set. This leads to the possibility of getting the wrong replica record on a disk set take. An error message is displayed, and user intervention is required.

Use the `-q` to query the disk set and the `-t`, `-u`, and `-y`, options to select the tag and take the disk set. See `OPTIONS`.

Mediator Configuration

SVM provides support for a low-end HA solution consisting of two hosts that share only two strings of drives. The hosts in this type of configuration, referred to as *mediators* or mediator hosts, run a special daemon, `rpc.metamedd(1M)`. The mediator hosts take on additional responsibilities to ensure that data is available in the case of host or drive failures.

A mediator configuration can survive the failure of a single host or a single string of drives, without administrative intervention. If both a host and a string of drives fail (multiple failures), the integrity of the data cannot be guaranteed. At this point, administrative intervention is required to make the data accessible. See `mediator(7D)` for further details.

Use the `-m` option to add or delete a mediator host. See `OPTIONS`.

オプション

The following options are supported:

`-a` *drivename*

Add drives or hosts to the named set. For a drive to be accepted into a set, the drive must not be in use within another metadevice or disk set, mounted on, or swapped on. When the drive is accepted into the set, it is repartitioned and the metadevice state database replica (for the set) can be placed on it. However, if a slice 7 (or slice 6 on an EFI labelled device), starts at cylinder 0, and is large enough to hold a state database replica, then the disk is not repartitioned. Also, a drive is not accepted if it cannot be found on all hosts specified as part of the set. This means that if a host within the specified set is unreachable due to network problems, or is administratively down, the add fails.

Specify a drive name in the form `c`*num*`t`*num*`d`*num*. Do not specify a slice number (`s`*num*). For drives in a Sun Cluster, you must specify a complete pathname for each drive. Such a name has the form:

/dev/did/[r]dsk/d*num*

-a | -d | -m *mediator_host_list*
Add (-a) or delete (-d) mediator hosts to the specified disk set. A *mediator_host_list* is the nodename (established by svc:/system/identity:node smf(5) service) of the mediator host to be added and (for adding) up to two other aliases for the mediator host. The nodename and aliases for each mediator host are separated only by commas. Up to three mediator hosts can be specified for the named disk set. Specify only the nodename of that host as the argument to -m to delete a mediator host.

In a single metaset command you can add or delete up to three mediator hosts. See EXAMPLES.

-A {enable | disable}
Specify auto-take status for a disk set. If auto-take is enabled for a set, the disk set is automatically taken at boot, and file systems on volumes within the disk set can be mounted through /etc/vfstab entries. Only a single host can be associated with an auto-take set, so attempts to add a second host to an auto-take set or attempts to configure a disk set with multiple hosts as auto-take fails with an error message. Disabling auto-take status for a specific disk set causes the disk set to revert to normal behavior. That is, the disk set is potentially shared (non-concurrently) among hosts, and unavailable for mounting through /etc/vfstab.

-b
Insure that the replicas are distributed according to the replica layout algorithm. This can be invoked at any time, and does nothing if the replicas are correctly distributed. In cases where the user has used the metadb command to manually remove or add replicas, this command can be used to insure that the distribution of replicas matches the replica layout algorithm.

-C {take | release | purge}
Do not interact with the Cluster Framework when used in a Sun Cluster 3 environment. In effect, this means do not modify the Cluster Configuration Repository. These options should only be used to fix a broken disk set configuration.

take
Take ownership of the disk set but do not inform the Cluster Framework that the disk set is available. This option is not for use with a multi-owner disk set.

release
Release ownership of the disk set without informing the Cluster Framework. This option should only be used if the disk set ownership was taken with the corresponding -C take option. This option is not for use with a multi-owner disk set.

purge
Remove the disk set without informing the Cluster Framework that the disk set has been purged. This option should only be used when the disk set is not accessible and requires rebuilding.

-d *drivename*

Delete drives or hosts from the named disk set. For a drive to be deleted, it must not be in use within the set. The last host cannot be deleted unless all of the drives within the set are deleted. Deleting the last host in a disk set destroys the disk set.

Specify a drive name in the form c*num*t*num*d*num*. Do not specify a slice number (s*num*). For drives in a Sun Cluster, you must specify a complete pathname for each drive. Such a name has the form:

```
/dev/did/[r]dsk/dnum
```

This option fails on a multi-owner disk set if attempting to withdraw the master node while other nodes are in the set.

-f

Force one of three actions to occur: takes ownership of a disk set when used with -t; deletes the last disk drive from the disk set; or deletes the last host from the disk set. Deleting the last drive or host from a disk set requires the -d option.

When used to forcibly take ownership of the disk set, this causes the disk set to be grabbed whether or not another host owns the set. All of the disks within the set are taken over (reserved) and fail fast is enabled, causing the other host to panic if it had disk set ownership. The metadevice state database is read in by the host performing the take, and the shared metadevices contained in the set are accessible.

You can use this option to delete the last drive in the disk set, because this drive would implicitly contain the last state database replica.

You can use -f option to delete hosts from a set. When specified with a partial list of hosts, it can be used for one-host administration. One-host administration could be useful when a host is known to be non-functional, thus avoiding timeouts and failed commands. When specified with a complete list of hosts, the set is completely deleted. It is generally specified with a complete list of hosts to clean up after one-host administration has been performed.

-h *hostname...*

Specify one or more host names to be added to or deleted from a disk set. Adding the first host creates the set. The last host cannot be deleted unless all of the drives within the set have been deleted. The host name is not accepted if all of the drives within the set cannot be found on the specified host. The host name is the nodename established by the svc:/system/identity:node smf(5) service.

-j

Join a host to the owner list for a multi-owner disk set. The concepts of take and release, used with traditional disk sets, do not apply to multi-owner sets, because multiple owners are allowed.

As a host boots and is brought online, it must go through three configuration levels to be able to use a multi-owner disk set:

1. It must be included in the cluster nodelist, which happens automatically in a cluster or single-node sitatuion.
2. It must be added to the multi-owner disk set with the `-a -h` options documented elsewhere in this man page
3. It must join the set. When the host is first added to the set, it is automatically joined.

On manual restarts, the administrator must manually issue

`metaset -s` *multinodesetname* `-j`

to join the host to the owner list. After the cluster reconfiguration, when the host reenters the cluster, the node is automatically joined to the set. The `metaset -j` command joins the host to all multi-owner sets that the host has been added to. In a single node situation, joining the node to the disk set starts any necessary resynchronizations.

`-L`
When adding a disk to a disk set, force the disk to be repartitioned using the standard Solaris Volume Manager algorithm. See `DESCRIPTION`.

`-l` *length*
Set the size (in blocks) for the metadevice state database replica. The length can only be set when adding a new drive; it cannot be changed on an existing drive. The default (and maximum) size is 8192 blocks, which should be appropriate for most configurations. Replica sizes of less than 128 blocks are not recommended.

`-M`
Specify that the disk set to be created or modified is a multi-owner disk set that supports multiple concurrent owners.

This option is required when creating a multi-owner disk set. Its use is optional on all other operations on a multi-owner disk set and has no effect. Existing disk sets cannot be converted to multi-owner sets.

`-o`
Return an exit status of `0` if the local host or the host specified with the `-h` option is the owner of the disk set.

`-P`
Purge the named disk set from the node on which the `metaset` command is run. The disk set must not be owned by the node that runs this command. If the node does own the disk set, the command fails.

If you need to delete a disk set but cannot take ownership of the set, use the `-P` option.

`-q`
Displays an enumerated list of tags pertaining to “tagged data" that can be encountered during a take of the ownership of a disk set.

This option is not for use with a multi-owner disk set.

`-r`

Release ownership of a disk set. All of the disks within the set are released. The metadevices set up within the set are no longer accessible.

This option is not for use with a multi-owner disk set.

`-s` *setname*

Specify the name of a disk set on which `metaset` works. If no *setname* is specified, all disk sets are returned.

`-t`

Take ownership of a disk set safely. If `metaset` finds that another host owns the set, this host is not be allowed to take ownership of the set. If the set is not owned by any other host, all the disks within the set are owned by the host on which `metaset` was executed. The metadevice state database is read in, and the shared metadevices contained in the set become accessible. The `-t` option takes a disk set that has stale databases. When the databases are stale, `metaset` exits with code 66, and prints a message. At that point, the only operations permitted are the addition and deletion of replicas. Once the addition or deletion of the replicas has been completed, the disk set should be released and retaken to gain full access to the data.

This option is not for use with a multi-owner disk set.

`-u` *tagnumber*

Once a tag has been selected, a subsequent take with `-u` *tagnumber* can be executed to select the data associated with the given *tagnumber*.

w

Withdraws a host from the owner list for a multi-owner disk set. The concepts of take and release, used with traditional disk sets, do not apply to multi-owner sets, because multiple owners are allowed.

Instead of releasing a set, a host can issue

```
metaset -s multinodesetname -w
```

to withdraw from the owner list. A host automatically withdraws on a reboot, but can be manually withdrawn if it should not be able to use the set, but should be able to rejoin at a later time. A host that withdrew due to a reboot can still appear joined from other hosts in the set until a reconfiguration cycle occurs.

`metaset -w` withdraws from ownership of all multi-owner sets of which the host is a member. This option fails if you attempt to withdraw the master node while other nodes are in the disk set owner list. This option cancels all resyncs running on the node. A cluster reconfiguration process that is removing a node from the cluster membership list effectively withdraws the host from the ownership list.

`-y`

Execute a subsequent take. If the take operation encounters "tagged data," the take operation exits with code 2. You can then run the `metaset` command with the `-q` option to see an enumerated list of tags.

使用例

例 1 Defining a Disk Set

This example defines a disk set.

```
# metaset -s relo-red -a -h red blue
```

The name of the disk set is `relo-red`. The names of the first and second hosts added to the set are `red` and `blue`, respectively. (The hostname is the nodename established by `svc:/system/identity:node` smf(5) service.) Adding the first host creates the disk set. A disk set can be created with just one host, with the second added later. The last host cannot be deleted until all of the drives within the set have been deleted.

例 2 Adding Drives to a Disk Set

This example adds drives to a disk set.

```
# metaset -s relo-red -a c2t0d0 c2t1d0 c2t2d0 c2t3d0 c2t4d0 c2t5d0
```

The name of the previously created disk set is `relo-red`. The names of the drives are `c2t0d0`, `c2t1d0`, `c2t2d0`, `c2t3d0`, `c2t4d0`, and `c2t5d0`. There is no slice identifier ("s*x*") at the end of the drive names.

例 3 Adding Multiple Mediator Hosts

The following command adds three mediator hosts to the specified disk set.

```
# metaset -s mydiskset -a -m myhost1,alias1 myhost2,alias2 myhost3,alias3
```

例 4 Purging a Disk Set from the Node

The following command purges the disk set `relo-red` from the node:

```
# metaset -s relo-red -P
```

例 5 Querying a Disk Set for Tagged Data

The following command queries the disk set `relo-red` for a list of the tagged data:

```
# metaset -s relo-red -q
```

This command produces the following results:

```
The following tag(s) were found:
 1 - vha-1000c - Fri Sep 20 17:20:08 2002
 2 - vha-1000c - Mon Sep 23 11:01:27 2002
```

例 6 Selecting a tag and taking a Disk set

The following command selects a tag and takes the disk set `relo-red`:

```
# metaset -s relo-red -t -u 2
```

例 7 Defining a Multi-Owner Disk Set

The following command defines a multi-owner disk set:

```
# metaset -s blue -M -a -h hahost1 hahost2
```

The name of the disk set is `blue`. The names of the first and second hosts added to the set are `hahost1` and `hahost2`, respectively. The hostname is the nodename established by `svc:/system/identity:node` `smf(5)` service. Adding the first host creates the multi-owner disk set. A disk set can be created with just one host, with additional hosts added later. The last host cannot be deleted until all of the drives within the set have been deleted.

ファイル

`/etc/lvm/md.tab`
: Contains list of metadevice configurations.

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metarename(1M)`, `metareplace(1M)`, `metassist(1M)`, `metastat(1M)`, `metasync(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `smf(5)`, `md(7D)`

『Solaris ボリュームマネージャの管理』

注意事項

Disk set administration, including the addition and deletion of hosts and drives, requires all hosts in the set to be accessible from the network.

名前

metassist – automated volume creation utility to support Solaris Volume Manager

形式

```
metassist -V

metassist -?

metassist create [-v n] [-c] -F config_file

metassist create [-v n] [-c | -d] -F request_file

metassist create [-v n] [-c | -d] [-f] [-n name]
     [-p datapaths] [-r redundancy]
     [-a available [,available,...]]
     [-u unavailable [,unavailable,...]] -s setname -S size

metassist create -?
```

機能説明

The `metassist` command provides assistance, through automation, with common Solaris Volume Manager tasks.

SUBCOMMANDS

The following subcommands are supported:

`create`

The `create` subcommand creates one or more Solaris Volume Manager volumes. You can specify this request on the command line or in a file specified on the command line.

If you create a volume using the command line, you can specify the characteristics of the volume in terms of the desired quality of service it will provide - its size, the number of redundant copies of the data it contains, the number of data paths by which it is accessible, and whether faulty components are replaced automatically. The diskset in which the volume will reside and the volume's size must be specified on the command line in this form of the command.

If you create a volume using a request in a file, you can specify the characteristics of the volume in terms of the quality of service they provide, as on the command line. Alternatively, the file can specify the types and component parts of the volume, (for example, mirrors, stripes, concatenations, and their component slices). The file may also specify volumes partly in terms of their types and partly in terms of their component parts, and may specify the characteristics of more than one volume. All volumes specified in a file must reside in the same diskset, whose name must be specified in the file.

If you specify the `-c` or `-d` option on the command line, the command runs without creating an actual volume or volumes. Instead, it outputs either a a Bourne shell command script (`-c` option) or a volume configuration (`-d` option). The command script, when run, creates the specified volume or volumes. The volume configuration specifies the volume or volumes in complete detail, naming all their components.

The input file given on the command line can take one of the following forms:

- a volume request, which specifies a request for a volume with explicit attributes and components, or matching a given quality of service
- a volume configuration, produced by a previous execution of the command

オプション

The following option is mandatory if you specify a volume request or volume configuration in a file:

`-F` *config_file* | *request_file*
: Specify the volume request or volume configuration file to process. If *config_file* or *request_file* is `-`, it is read from standard input.

  The `-d` option cannot be specified when *inputfile* is a volume configuration file.

The following options are mandatory if you specify a volume request on the command line:

`-s` *set*
: Specify the disk set to use when creating volumes. All the volumes and hot spare pools are created in this disk set. If necessary, disks are moved into the diskset for use in the volumes and hot spare pools. If the diskset doesn't exist the command creates it. This option is required. `metassist` works entirely within a named disk set. Use of the local, or unnamed disk set, is not allowed.

`-S` *size*
: Specify the size of the volume to be created. The size argument consists of a numeric value (a decimal can be specified) followed by KB, MB, GB, or TB, indicating kilobytes, megabytes, gigabytes, or terabytes, respectively. Case is ignored when interpreting this option. This option is required.

The following options are optional command line parameters:

`-a` *device1*`,`*device2*`,...`
: Explicitly specify the devices that can be used in the creation of this volume. Named devices may be controllers or disks. Only used when specifying a volume on the command line.

`-c`
: Output the command script that would implement the specified or generated volume configuration. The command script is not run, and processing stops at this stage.

`-d`
: Output the volume configuration that satisfies the specified or generated volume request. No command script is generated or executed, and processing stops at this stage.

`-f`
: Specify whether the volume should support automatic component replacement after a fault. If this option is specified, a mirror is created and its submirrors are associated with a hot spare.

`-n` *name*
: Specify the name of the new volume. See `metainit(1M)` for naming guidelines.

`-p` *n*	Specify the number of required paths to the storage volume. The value of *n* cannot be greater than the number of different physical paths and logical paths to attached storage. Only used when specifying a volume on the command line.
`-r` *n*	Specify the redundancy level (`0`-`4`) of the data. The default is `0`. Only used when specifying a volume on the command line. If redundancy is `0`, a stripe is created. If redundancy is `1` or greater, a mirror with this number of submirrors is created. In this case, the volume can suffer a disk failure on *n*`-1` copies without data loss. With the use of hot spares (see the `-f` option), a volume can suffer a disk failure on *n*+`hsps-1` volumes without data loss, assuming non-concurrent failures.
`-u` *device1*`,`*device2*`,...`	Explicitly specify devices to exclude in the creation of this volume. Named devices can be controllers or disks. You can use this option alone, or to exclude some of the devices listed as available with the `-a` option, Only used when specifying a volume on the command line.
`-v` *value*	Specify the level of verbosity. Values from `0` to `2` are available, with higher numbers specifying more verbose output when the command is run. `-v 0` indicates silent output, except for errors or other critical messages. The default level is `1`.
`-V`	Display program version information.
`-?`	Display help information. This option can follow a subcommand for subcommand-specific help.

使用例

例 1 Creating a Mirror

The following example creates a two-way, 36Gb mirror on available devices from controller 1 and controller 2. It places the volume in diskset `mirrorset`.

```
# metassist create -r 2 -a c1,c2 -s mirrorset -S 36GB
```

例 2 Creating a Mirror with Additional Fault Tolerance

The following example creates a two-way, 36Gb mirror on available devices from controller 1 and controller 2. It provides additional fault tolerance in the form of a hot spare. It places the volume in diskset `mirrorset`.

```
# metassist create -f -r 2 -a c1,c2 -s mirrorset -S 36GB
```

例 3 Creating a Three-way Mirror and Excluding Devices

The following example creates a three-way, 180Gb mirror from storage devices on controller 1 or controller 2. It excludes the disks `c1t2d0` and `c2t2d1` from the volume. It places the volume in diskset `mirrorset`.

```
metassist create -r 3 -a c1,c2 -u c1t2d0, c2t2d1 \
    -s mirrorset -S 180GB
```

例 4 Determining and Implementing a Configuration

The following example determines and implements a configuration satisfying the request specified in a request file:

```
# metassist create -F request.xml
```

例 5 Determining a Configuration and Saving It in a volume-config File

The following example determines a configuration which satisfies the given request. It saves the configuration in a volume-config file without implementing it:

```
# metassist create -d -F request.xml > volume-config
```

例 6 Determining a Configuration and Saving It in a Shell Script

The following example determines a configuration which satisfies the given request. It saves the configuration in a shell script without implementing it:

```
# metassist create -c -F request.xml > setupvols.sh
```

例 7 Implementing the Given volume-config

The following example implements the given volume-config:

```
# metassist create -F config.xml
```

例 8 Converting the Given volume-config to a Shell Script

The following example converts the given volume-config to a shell script that you can run later:

```
# metassist create -c -F config.xml > setupvols.sh
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

- `/usr/share/lib/xml/dtd/volume-request.dtd`
- `/usr/share/lib/xml/dtd/volume-defaults.dtd`
- `/usr/share/lib/xml/dtd/volume-config.dtd`

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metastat(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), volume-config(4), volume-request(4), attributes(5), md(7D)

注意事項

The quality of service arguments are mutually exclusive with the -F *inputfile* argument.

When specifying a request file or quality of service arguments on the command line, the /etc/default/metassist.xml file is read for global and per-disk set defaults.

Characteristics of this file are specified in the DTD, in /usr/share/lib/xml/dtd/volume-defaults.dtd.

Characteristics of the XML request file are specified in the DTD, in /usr/share/lib/xml/dtd/volume-request.dtd.

Characteristics of the XML configuration file are specified in the DTD, in /usr/share/lib/xml/dtd/volume-config.dtd.

This command must be run as root.

This command requires a functional Solaris Volume Manager configuration before it runs.

名前

metastat – display status for metadevice or hot spare pool

形式

```
/usr/sbin/metastat -h
```

```
/usr/sbin/metastat [-a] [-B] [-D] [-c] [-i] [-p] [-q]
     [-s setname] [-t] [metadevice...] [hot_spare_pool...]
```

```
/usr/sbin/metastat [-a] [-B] [-D] [-c] [-i] [-p] [-q]
     [-s setname] component...
```

機能説明

The `metastat` command displays the current status for each metadevice (including stripes, concatenations, concatenations of stripes, mirrors, RAID5, soft partitions, and trans devices) or hot spare pool, or of specified metadevices, components, or hot spare pools.

It is helpful to run the `metastat` command after using the `metattach` command to view the status of the metadevice.

`metastat` displays the state of each Solaris Volume Manager RAID-1 volume on the system. The possible states include:

`Okay`	The device reports no errors.
`Needs maintenance`	A problem has been detected. This requires that the system administrator replace the failed physical device. Volumes displaying `Needs maintenance` have incurred no data loss, although additional failures could risk data loss. Take action as quickly as possible.
`Last erred`	A problem has been detected. Data loss is a possibility. This might occur if a component of a submirror fails and is not replaced by a hot spare, therefore going into `Needs maintenance` state. If the corresponding component also fails, it would go into `Last erred` state and, as there is no remaining valid data source, data loss could be a possibility.
`Unavailable`	A device cannot be accessed, but has not incurred errors. This might occur if a physical device has been removed with Solaris Dynamic Reconfiguration (DR) features, thus leaving the Solaris Volume Manager volume unavailable. It could also occur if an array or disk is powered off at system initialization, or if a >1TB volume is present when the system is booted in 32-bit mode. After the storage has been made available, run the `metastat` command with the `-i` option to update the status of the metadevices. This clears the unavailable state for accessible devices.

See the 『Solaris ボリュームマネージャの管理』 for instructions on replacing disks and handling volumes in Needs maintenance or Last erred states.

オプション

The following options are supported:

`-a` Display all disk sets. Only metadevices in disk sets that are owned by the current host are displayed.

`-B` Display the current status of all of the 64-bit metadevices and hot spares.

`-c` Display concise output.

There is one line of output for each metadevice. The output shows the basic structure and the error status, if any, for each metadevice.

The `-c` output format is distinct from the `-p` output format. The `-p` option does not display metadevice status and is not intended as human-readable output.

`-D` Display the current status of all of the descriptive name metadevices and hotspares.

`-h` Display usage message.

`-i` Check the status of RAID-1 (mirror) volumes, RAID-5 volumes, and hot spares. The inquiry checks each metadevice for accessibility, starting at the top level metadevice. When problems are discovered, the metadevice state databases are updated as if an error had occurred.

`-p` Display the list of active metadevices and hot spare pools in the same format as `md.tab`. See `md.tab(4)`.

The `-p` output is designed for snapshotting the configuration for later recovery or setup.

`-q` Display the status for metadevices without the device relocation information.

`-s` *setname* Specify the name of the disk set on which `metastat` works. Using the `-s` option causes the command to perform its administrative function within the specified disk set. Without this option, the command performs its function on metadevices and hot spare pools in the local disk set.

`-t` Display the current status and timestamp for the specified metadevices and hot spare pools. The timestamp provides the date and time of the last state change.

オペランド

The following operands are supported:

component Display the status of the component hosting a soft partition, including extents, starting blocks, and block count.

hot_spare_pool Display the status of the specified hot spare pool(s).

metadevice Display the status of the specified metadevice(s). If a trans metadevice is specified, the status of the master and log devices is also displayed. Trans metadevices have been replaced by UFS logging. See NOTES.

使用例

例 1 Output Showing Mirror with Two Submirrors

The following example shows the partial output of the metastat command after creating a mirror, opt_mirror, consisting of two submirrors, opt_sub1 and opt_sub2.

```
# metastat opt_mirror
opt_mirror: Mirror
    Submirror 0: opt_sub1
      State: Okay
    Submirror 1: opt_sub2
      State: Resyncing
    Resync in progress: 15 % done
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 2006130 blocks
    .
    .
    .
```

例 2 Soft Partition on Mirror with Submirror

The following example shows the partial output of the metastat command after creating a soft partition, d3, on concat d2, which is built on a soft partition.

```
# metastat
d2: Concat/Stripe
    Size: 204800 blocks
    Stripe 0:
        Device              Start Block  Dbase State        Hot Spare
        d0                         0     No    Okay

d0: Soft Partition
    Component: c0t3d0s0
    Status: Okay
    Size: 204800 blocks
        Extent              Start Block  Block count
             0                      129       204800

d3: Soft Partition
    Component: d2
    Status: Okay
    Size: 202752 blocks
         Extent              Start Block  Block count
              0                      129       202752
```

例 3 Trans Metadevice

The following example shows the output of the `metastat` command after creating a trans metadevice.

```
# metastat
d2: Concat/Stripe
    Size: 204800 blocks
    Stripe 0:
        Device              Start Block  Dbase State        Hot Spare
        d0                         0     No    Okay

d0: Soft Partition
    Component: c0t3d0s0
    Status: Okay
    Size: 204800 blocks
        Extent              Start Block  Block count
             0                     129       204800

d3: Soft Partition
    Component: d2
    Status: Okay
    Size: 202752 blocks
         Extent              Start Block  Block count
              0                     129       202752
```

例 4 Multi-owner disk set

The following example shows the output of the `metastat` command with a multi-owner disk set and application-based mirror resynchronization option. Application-based resynchronization is set automatically if needed.

```
# metastat -s oban
oban/d100: Mirror
    Submirror 0: oban/d10
      State: Okay
    Submirror 1: oban/d11
      State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Resync option: application based
    Owner: None
    Size: 1027216 blocks (501 MB)

oban/d10: Submirror of oban/d100
    State: Okay
    Size: 1027216 blocks (501 MB)
    Stripe 0:
```

例 4 Multi-owner disk set (続き)

```
        Device     Start Block  Dbase     State Reloc Hot Spare
        c1t3d0s0          0     No         Okay

oban/d11: Submirror of oban/d100
    State: Okay
    Size: 1027216 blocks (501 MB)
    Stripe 0:
        Device     Start Block  Dbase     State Reloc Hot Spare
        c1t4d0s0          0     No         Okay
```

警告

metastat displays states as of the time the command is entered. It is unwise to use the output of the metastat -p command to create a md.tab(4) file for a number of reasons:

- The output of metastat -p might show hot spares being used.
- It might show mirrors with multiple submirrors. See metainit(1M) for instructions for creating multi-way mirrors using metainit and metattach.
- A slice may go into an error state after metastat -p is issued.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metadetach(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metasync(1M), metattach(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

注意事項

Trans metadevices have been replaced by UFS logging. Existing trans devices are *not* logging--they pass data directly through to the underlying device. See mount_ufs(1M) for more information about UFS logging.

名前

metasync – handle metadevice resync during reboot

形式

```
/usr/sbin/metasync -h
/usr/sbin/metasync [-s setname] [buffer_size] metadevice
/usr/sbin/metasync [-s setname] -r [buffer_size]
/usr/sbin/metasync -p metadevice
/usr/sbin/metasync -c metadevice
```

機能説明

The `metasync` command starts a resync operation on the specified *metadevice*. All components that need to be resynced are resynced. If the system crashes during a RAID5 initialization, or during a RAID5 resync, either an initialization or resync restarts when the system reboots.

Applications are free to access a metadevice at the same time that it is being resynced by `metasync`. Also, `metasync` performs the copy operations from inside the kernel, which makes the utility more efficient.

Use the `-r` option in boot scripts to resync all possible submirrors.

オプション

The following options are supported:

`-c` *metadevice*
: Cancels the resync that is in progress on the specified metadevice. The resync will be stopped at its current point and can be resumed by running the "`metasync metadevice`" command. This option only applies to RAID1 volumes.

`-h`
: Displays usage message.

`-p` *metadevice*
: Regenerates parity information for RAID5 metadevices.

`-s` *setname*
: Specifies the name of the diskset on which `metasync` will work. Using the `-s` option will cause the command to perform its administrative function within the specified diskset. Without this option, the command will perform its function on local metadevices.

`-r`
: Specifies that the `metasync` command handle special resync requirements during a system reboot. `metasync -r` should only be invoked from the `svc:/system/mdmonitor` service. The `metasync` command only resyncs those metadevices that need to be resynced. `metasync` schedules all the mirror resyncs according to their pass numbers.

  To override the default buffer_size value used by the `svc:/system/mdmonitor` service, you can edit `/etc/system` to specify:

  ```
  set md_mirror:md_resync_bufsz = 2048
  ```

  so that resyncs occur as quickly as possible.

オペランド

buffer_size Specifies the size (number of 512-byte disk blocks) of the internal copy buffer for the mirror resync. The size defaults to 1024 512-byte disk blocks. It can be no more than 2048 blocks. For best performance (quickest completion of the resync), 2048 blocks is the recommended size.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm

関連項目

`mdmonitord(1M)`, `metaclear(1M)`, `metadb(1M)`, `metadetach(1M)`, `metahs(1M)`, `metainit(1M)`, `metaoffline(1M)`, `metaonline(1M)`, `metaparam(1M)`, `metarecover(1M)`, `metarename(1M)`, `metareplace(1M)`, `metaset(1M)`, `metassist(1M)`, `metastat(1M)`, `metattach(1M)`, `md.tab(4)`, `md.cf(4)`, `mddb.cf(4)`, `md.tab(4)`, `attributes(5)`, `md(7D)`

『Solaris ボリュームマネージャの管理』

注意事項

The `metasync` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/mdmonitor
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

metattach, metadetach – attach or detach a metadevice

形式

```
/usr/sbin/metattach [-h]
/usr/sbin/metattach [-s setname] mirror [metadevice]
/usr/sbin/metattach [-s setname] [-i interlace] concat/stripe component...
/usr/sbin/metattach [-s setname] RAID component...
/usr/sbin/metattach [-s setname] [-A alignment] softpart size | all
/usr/sbin/metadetach [-s setname] [-f] mirror submirror
/usr/sbin/metadetach [-s setname] [-f] trans
```

機能説明

`metattach` adds submirrors to a mirror, grows metadevices, or grows soft partitions. Growing metadevices can be done without interrupting service. To grow the size of a mirror or trans, the slices must be added to the submirrors or to the master devices.

Solaris Volume Manager supports storage devices and logical volumes greater than 1 terabyte (TB) when a system runs a 64-bit Solaris kernel. Support for large volumes is automatic. If a device greater than 1 TB is created, Solaris Volume Manager configures it appropriately and without user intervention.

If a system with large volumes is rebooted under a 32–bit Solaris kernel, the large volumes are visible through `metastat` output. Large volumes cannot be accessed, modified or deleted, and no new large volumes can be created. Any volumes or file systems on a large volume in this situation are also unavailable. If a system with large volumes is rebooted under a version of Solaris prior to the Solaris 9 4/03 release, Solaris Volume Manager does not start. You must remove all large volumes before Solaris Volume Manager runs under an earlier version of the Solaris Operating System.

Solaris Volume Manager supports one-to-four-way mirrors. You can only attach a metadevice to a mirror if there are three or fewer submirrors beneath the mirror. Once a new metadevice is attached to a mirror, `metattach` automatically starts a resync operation to the new submirror.

`metadetach` detaches submirrors from mirrors and logging devices from trans metadevices.

When a submirror is detached from a mirror, it is no longer part of the mirror, thus reads and writes to and from that metadevice by way of the mirror are no longer performed through the mirror. Detaching the only existing submirror is not allowed. Detaching a submirror that has slices reported as needing maintenance (by `metastat`) is not allowed unless the `-f` (force) flag is used.

`metadetach` also detaches the logging device from a trans. This step is necessary before you can clear the trans volume. Trans metadevices have been replaced by UFS logging. Existing trans devices are not logging. They pass data directly through to the underlying device. See `mount_ufs(1M)` for more information about UFS logging.

Detaching the logging device from a busy trans device is not allowed unless the `-f` (force) flag is used. Even so, the logging device is not actually detached until the trans is idle. The trans is in the *Detaching* state (`metastat`) until the logging device is detached.

オプション

Root privileges are required for all of the following options except `-h`.

The following options are supported:

`-A` *alignment*
: Set the value of the soft partition extent alignment. Use this option when it is important specify a starting offset for the soft partition. It preserves the data alignment between the metadevice address space and the address space of the underlying physical device.

  For example, a hardware device that does checksumming should not have its I/O requests divided by Solaris Volume Manager. In this case, use a value from the hardware configuration as the value for the alignment. When using this option in conjunction with a software I/O load, the alignment value corresponds to the I/O load of the application. This prevents I/O from being divided unnecessarily and affecting performance.

`-f`
: Force the detaching of metadevices that have components that need maintenance or are busy. You can use this option only when a mirror is in a maintenance state that can be fixed with `metareplace(1M)`. If the mirror is in a maintenance state that can only be fixed with `metasync(1M)` (as shown by the output of `metastat(1M)`), `metadetach -f` has no effect, because the mirrors must be resynchronized before one of them can be detached.

`-h`
: Display a usage message.

`-i` *interlace*
: Specify the interlace value for stripes, where `size` is a specified value followed by either `k` for kilobytes, `m` for megabytes, or `b` for blocks. The units can be either uppercase or lowercase. If *size* is not specified, the size defaults to the interlace size of the last stripe of the metadevice. When an interlace size change is made on a stripe, it is carried forward on all stripes that follow.

`-s` *setname*
: Specify the name of the diskset on which the `metattach` command or the `metadetach` command works.. Using the `-s` option causes the command to perform its administrative function within the specified diskset. Without this option, the command performs its function on local metadevices.

オペランド

The following operands are supported:

component
: The logical name for the physical slice (partition) on a disk drive, such as `/dev/dsk/c0t0d0s2`, being added to the concatenation, stripe, concatenation of stripes, or RAID5 metadevice.

concat/stripe
: The metadevice name of the concatenation, stripe, or concatenation of stripes.

log	The metadevice name of the logging device to be attached to the trans metadevice.
metadevice	The metadevice name to be attached to the mirror as a submirror. This metadevice must have been previously created by the `metainit` command.
mirror	The name of the mirror.
RAID	The metadevice name of the RAID5 metadevice.
size | *all*	The amount of space to add to the soft partition in `K` or `k` for kilobytes, `M` or `m` for megabytes, `G` or `g` for gigabytes, `T` or `t` for terabytes, and `B` or `b` for blocks (sectors). All values represent powers of 2, and upper and lower case options are equivalent. Only integer values are permitted. The literal `all` specifies that the soft partition should grow to occupy all available space on the underlying volume.
softpart	The metadevice name of the existing soft partition.
submirror	The metadevice name of the submirror to be detached from the mirror.
trans	The metadevice name of the trans metadevice (not the master or logging device).

使用例

例 1 Concatenating a New Slice to a Metadevice

This example concatenates a single new slice to an existing metadevice, `Volume.1`. Afterwards, you would use the `growfs(1M)` command to expand the file system.

```
# metattach Volume.1 /dev/dsk/c0t1d0s2
```

例 2 Detaching Logging Device from Trans Metadevice

This example detaches the logging device from a trans metadevice `d9`. Notice that you do not have to specify the logging device itself, as there can only be one.

```
# metadetach d9
```

例 3 Expanding a RAID5 Metadevice

This example expands a RAID5 metadevice, `d45`, by attaching another slice.

```
# metattach d45 /dev/dsk/c3t0d0s2
```

When you add additional slices to a RAID5 metadevice, the additional space is devoted to data. No new parity blocks are allocated. The data on the added slices is, however, included in the overall parity calculations, so it is protected against single-device failure.

例 4 Expanding a Soft Partition

The following example expands a soft partition, `d42`, attaching all space available on the underlying device.

例 4 Expanding a Soft Partition (続き)

```
# metattach d42 all
```

When you add additional space to a soft partition, the additional space is taken from any available space on the slice and might not be contiguous with the existing soft partition.

例 5 Adding Space to Two-Way Mirror

This example adds space to a two-way mirror by adding a slice to each submirror. Afterwards, you would use the `growfs(1M)` command to expand the file system.

```
# metattach d9 /dev/dsk/c0t2d0s5
# metattach d10 /dev/dsk/c0t3d0s5
```

This example tells the mirror to grow to the size of the underlying devices

```
# metattach d11
```

This example increases the size of the UFS on the device so the space can be used.

```
# growfs -M /export /dev/md/rdsk/d11
```

例 6 Detaching a Submirror from a Mirror

This example detaches a submirror, `d2`, from a mirror, `d4`.

```
# metadetach d4 d2
```

例 7 Adding Four Slices to Metadevice

This example adds four slices to an existing metadevice, `d9`. Afterwards, you would use the `growfs(1M)` command to expand the file system.

```
# metattach d9 /dev/dsk/c0t1d0s2 /dev/dsk/c0t2d0s2 \\
     /dev/dsk/c0t3d0s2 /dev/dsk/c0t4d0s2
```

例 8 Setting the Value of the Soft Partition Extent Alignment

This example shows how to set the alignment of the soft partition to 1mb when the soft partition is expanded.

```
# metattach -s red -A 2m d13 1m
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

mdmonitord(1M), metaclear(1M), metadb(1M), metahs(1M), metainit(1M), metaoffline(1M), metaonline(1M), metaparam(1M), metarecover(1M), metarename(1M), metareplace(1M), metaset(1M), metassist(1M), metastat(1M), metasync(1M), md.tab(4), md.cf(4), mddb.cf(4), md.tab(4), attributes(5), md(7D)

『Solaris ボリュームマネージャの管理』

警告

This section provides information regarding warnings for devices greater than 1 TB and for multi-way mirrors.

Devices and Volumes Greater Than 1 TB

Do not create large (>1 TB) volumes if you expect to run the Solaris Operating System with a 32-bit kernel or if you expect to use a version of the Solaris Operating System prior to Solaris 9 4/03.

Multi-Way Mirrors

When a submirror is detached from its mirror, the data on the metadevice might not be the same as the data that existed on the mirror prior to running metadetach. In particular, if the -f option was needed, the metadevice and mirror probably do not contain the same data.

注意事項

Trans metadevices have been replaced by UFS logging. Existing trans devices are *not* logging. They pass data directly through to the underlying device. See mount_ufs(1M) for more information about UFS logging.

名前

mib2mof – generate MOF file(s) from input SNMP MIB file(s)

形式

```
/usr/sadm/bin/mib2mof [-n] [-d directory] [-q] [-c] [-a]
     [-h] files
```

機能説明

The `mib2mof` utility reads input Management Information Base (MIB) files and produces one or more Managed Object Format (MOF) files. MOF files contain a Common Information Model (CIM) class declaration that represents the MIB for the Solaris Simple Network Management Protocol (SNMP) provider. The SNMP provider allows Web-Based Enterprise Management (WBEM) applications to access SNMP device information.

SNMP scalar variables map to properties in the CIM class. Qualifiers on each property convey the following MIB information for each scalar variable:

- syntax
- read/write access
- OID (Object IDentifier)
- description (optional)
- index (if the variable is within a group [sequence] that defines a row)

The syntax of an SNMP scalar variable is represented in a CIM class by the property's CIM datatype. All properties are marked with write access (true or false).

The following table shows how a Solaris SNMP datatype in a MIB maps to a Web-Based Enterprise Management (WBEM) CIM datatype and then to an SNMP datatype used by the WBEM SNMP API:

SNMP SMI Datatype	SNMP Ver.	CIM Datatype	SNMP API Object type
INTEGER	v1	sint32	SnmpInt
OCTET STRING	v1	string	SnmpString
OBJECT IDENTIFIER	v1	string	SnmpOid
IpAddress	v1	string	SnmpIpAddress
Counter	v1	uint32	SnmpCounter
Gauge	v1	uint32	SnmpGauge
TimeTicks	v1	uint32	SnmpTimeticks
Opaque	v1	sint8[]	SnmpOpaque
DisplayString - see OCTET STRING	v1		
NetworkAddress - see IpAddress	v1		
Counter32 - see Counter	v2		
Counter64	v2	uint64	SnmpCounter64
Integer32	v2	sint32	SnmpInt
Gauge32 - see Gauge	v2		
Unsigned32	v2	uint32	SnmpGauge
TruthValue	v2	sint32	SnmpInt
BITS - see OCTET STRING	v2		

The `mib2mof` utility includes its required `Solaris_SNMPmib_core.txt` file (containing core `MIB` definitions), installed in `/usr/sadm/mof`. The `mib2mof` utility looks first for `mib` core file in local directory. If this file is not found in the local directory, `mib2mof` looks in `/usr/sadm/mof`.

A MOF file is generated for each SNMP group and table row sequence (that is, the columns in one row) found in the supplied MIBs. (This does not include the core MIB definitions contained in the `Solaris_SNMPmib_core.txt file`.)

There is no MOF file or property for an SNMP table - all table access is through the rows and columns of the table, and the SNMP variable for the table is marked as inaccessible in the MIB.

The MOF file created contains a CIM class that represents an SNMP group or row and a CIM class to represent a CIM association. The output file name (and CIM class) is of the format <SNMP_><MIB name><Group name>.mof.

オプション

The following options are supported:

`-a` Generate MOF files for all of the input MIB files. If `-a` is not given, a MOF file is generated only for the last file of the input list.

`-c` Do not use the default `Solaris_SNMPmib_core.txt` definitions file shipped with the Solaris SNMP Provider for WBEM. If this option is specified, you must specify another `MIB_CORE` definitions file as one of the input files.

`-d` *directory* Generate output MOF files in the specified directory.

`-h` Show how to invoke `mib2mof` and list its arguments.

`-n` Parse the input MIB files without generating any output.

`-q` Include the DESCRIPTION clause of `SNMP OBJECT-TYPE` as a qualifier in the generated MOF file.

オペランド

The following operands are supported:

files List of SNMP MIB files to be converted.

終了ステータス

The `mib2mof` utility terminates with exit status `0`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

`init.wbem(1M)`, `mofcomp(1M)`, `wbemadmin(1M)`, `attributes(5)`

名前

mibiisa – Sun SNMP Agent

形式

```
mibiisa [-ar] [-c config-dir] [-d debug-level] [-p port]
      [-t cache-timer]
```

機能説明

The `mibiisa` utility is an RFC 1157-compliant SNMP agent. It supports MIB-II as defined in RFC 1213, with Sun extensions under Sun's enterprise number. The MIB (Management Information Base) is both readable and writable. The `mibiisa` utility supports all SNMP protocol operations including GET-REQUEST, GETNEXT-REQUEST, SET-REQUEST, GET-REPLY, and TRAP.

The SMA (Systems Management Agent) is the default SNMP agent in Solaris. MIB-II subagent `mibiisa` does not run by default. To enable `mibiisa`, rename the configuration file from `/etc/snmp/conf/mibiisa.rsrc-` to `/etc/snmp/conf/mibiisa.rsrc`. SMA has the capability to handle any MIB-II requests.

The `mibiisa` utility supports the coldStart, linkUp, linkDown, and authentication traps. The authentication trap may be disabled by a command-line switch, which itself may be overridden by a management station writing to a MIB variable in the standard SNMP MIB group.

The `mibiisa` utility supports four distinct views of the MIB. The view used for any request is determined by the community string contained in that request.

To enhance security, `mibiisa` supports an option to block all writes to the MIB. You can also limit the set of management stations from which the agent will accept requests in the configuration file used when starting the `mibiisa`. See the セキュリティー section for more information.

Unless overridden, `mibiisa` uses UDP port 161, the standard SNMP port. The `mibiisa` utility issues traps through the same port on which it receives SNMP requests.

The `mibiisa` utility must run with super-user privileges and is typically started at system startup via `/etc/rc3.d`. `mibiisa` may not be started using `inetd(1M)`. When started, `mibiisa` detaches itself from the keyboard, disables all signals except `SIGKILL`, `SIGILL`, `SIGUSR1`, and `SIGUSR2`, and places itself in the background.

オプション

The following options are supported by `mibiisa`:

`-a`	Disable the generation of authentication traps. However, an SNMP manager may write a value into `snmpEnableAuthenTraps` to enable or disable authentication traps.
`-c` *config-dir*	Specify a directory where it expects `snmpd.conf` file, on startup. The default directory is `/etc/snmp/conf`.
`-d` *debug-level*	Debug. A value of 0 disables all debug and is the default. Levels 1 through 3 represent increasing levels of debug output. When `mibiisa` receives the

signal SIGUSR1, it resets the debug-level to 0. When mibiisa receives the signal SIGUSR2, it increments the debug-level by one.

Debug output is sent to the standard output in effect at the time mibiisa is started. No matter what debug level is in effect, certain significant events are logged in the system log.

-p *port* Define an alternative UDP port on which mibiisa listens for incoming requests. The default is UDP port 161.

-r Place the MIB into read-only mode.

-t *cache-timer* By default, information fetched from the kernel is considered to be valid for 45 seconds from the time it is retrieved. This cache lifetime may be altered with this parameter. You cannot set *cache-timer* to any value less than 1.

Configuration File

The snmpd.conf file is used for configuration information. Each entry in the file consists of a keyword followed by a parameter string. The keyword must begin in the first position. Parameters are separated from the keyword and from one another by white space. Case in keywords is ignored. Each entry must be contained on a single line. All text following (and including) a pound sign (#) is ignored. Keywords currently supported are:

sysdescr The value to be used to answer queries for sysDescr.

syscontact The value to be used to answer queries for sysContact.

syslocation The value to be used to answer queries for sysLocation.

trap The parameter names one or more hosts to receive traps. Only five hosts may be listed.

system-group-read-community The community name to get read access to the system group and Sun's extended system group.

system-group-write-community The community name to get write access to the system group and Sun's extended system group.

read-community The community name to get read access to the entire MIB.

write-community The community name to get write access to the entire MIB (implies read access).

trap-community The community name to be used in traps.

kernel-file The name of the file to use for kernel symbols.

managers The names of hosts that may send SNMP queries. Only five hosts may be listed on any one line. This keyword may be repeated for a total of 32 hosts.

`newdevice` The additional devices which are not built in SNMPD. The format is as follows: *newdevice* `type` *speed name* where *newdevice* is the keyword, `type` is an integer which has to match your schema file, *speed* is the new device's speed, and *name* is this new device's name.

An example `snmpd.conf` file is shown below:

```
sysdescr        Sun SNMP Agent, Sun Fire 4800, Company
                   Property Number 123456
syscontact      Cliff Claven
sysLocation     Room 1515, building 1
#
system-group-read-community     public
system-group-write-community    private
#
read-community  all_public
write-community all_private
#
trap            localhost
trap-community  SNMP-trap
#
#kernel-file    /vmunix
#
managers        lvs golden
managers        swap
```

Installation

The `mibiisa` utility and its configuration file, `snmpd.conf`, may be placed in any directory. However for Solaris 2.4 and subseqent releases, use `/usr/lib/snmp` for `mibiisa` itself and `/etc/snmp/conf` for the configuration file. You can modify the configuration file as appropriate. If you make any changes to `snmpd.conf` file keyword values, you must kill and restart `mibiisa` for the changes to take effect.

Your `/etc/services` file (or NIS equivalent) should contain the following entries:

`snmp`	`161/udp`		`# Simple Network Mgmt Protocol`
`snmp-trap`	`162/udp`	`snmptrap`	`# SNMP trap (event) messages`

The following is an example for Solaris 2.*x* and releases compatible with Solaris 2.*x*, such as Solaris 9:

```
#
# Start the SNMP agent
#
```

```
if [ -f /etc/snmp/conf/snmpd.conf -a -x
   /usr/lib/snmp/mibiisa ];
then
/opt/SUNWconn/snm/agents/snmpd
echo 'Starting SNMP-agent.'
```

セキュリティー

SNMP, as presently defined, offers relatively little security. The `mibiisa` utility accepts requests from other machines, which can have the effect of disabling the network capabilities of your computer. To limit the risk, the configuration file lets you specify a list of up to 32 manager stations from which `mibiisa` will accept requests. If you do not specify any such manager stations, `mibiisa` accepts requests from anywhere.

The `mibiisa` utility also allows you to mark the MIB as "read-only" by using the `-r` option.

`mibiisa` supports four different community strings. These strings, however, are visible in the configuration file and within the SNMP packets as they flow on the network.

The configuration file should be owned by, and readable only by super-user. In other words the mode should be:

```
-rw-------  1 root         2090 Oct 17 15:04 /etc/snmp/conf/snmpd.conf
```

Managers can be restricted based on the community strings. This can be configured by creating an optional secondary configuration file `/etc/snmp/conf/mibiisa.acl`. To enable such a restriction, add the security line in the `/etc/snmp/conf/mibiisa.rsrc` file.

An example `mibiisa.acl` file is as follows:

```
acl = {
        {
                communities = public
                access = read-only
                managers = xyz
        }
        {
                communities = private
                access = read-write
                managers = abc,pqrs
        }
}
```

An example `mibiisa.rsrc` file is as follows:

```
resource =
{
        {
                registration_file = "/etc/snmp/conf/mibiisa.reg"
                security = "/etc/snmp/conf/mibiisa.acl"
                policy = "spawn"
                type = "legacy"
```

```
                command = "/usr/lib/snmp/mibiisa -r -p $PORT"
        }
}
```

Mib

This section discusses some of the differences between the `mibiisa` MIB and the standard MIB-II (as defined in RFC 1213).

The following variables are read-only in the `mibiisa` MIB:

```
sysName
atIfIndex
ipDefaultTTL
```

These variables are read-write in the standard MIB-II.

The `mibiisa` MIB Address Translation tables support limited write access: only atPhysAddress may be written, either to change the physical address of an existing entry or to delete an entire ARP table entry.

The `mibiisa` MIB IP Net to Media table supports limited write access: only ipNetToMediaPhysAddress and ipNetToMediaType may be written, either to change the physical address of an existing entry or to delete an entire ARP table entry.

The following variables are read-write in the `mibiisa` MIB; however, these variables have fixed values. Any new values "set" to them are accepted, but have no effect:

```
ipRoutIfIndex
ipRouteMetric1
ipRouteMetric2
ipRouteMetric3
ipRouteMetric4
ipRouteType
ipRouteAge
ipRouteMask
ipRouteMetric5
```

The following `mibiisa` MIB variable reflects the actual state of the related table entry. "Sets" are accepted but have no effect:

```
tcpConnState
```

The following `mibiisa` MIB variables are readable, but return a fixed value:

`icmpInDestUnreachs`	Returns 1
`icmpInTimeExcds`	Returns 1
`icmpInParmProbs`	Returns 1
`icmpInSrcQuenchs`	Returns 1

icmpInRedirects	Returns 1
icmpInEchos	Returns 1
icmpInEchoReps	Returns 1
icmpInTimestamps	Returns 1
icmpInTimestampReps	Returns 1
icmpInAddrMasks	Returns 1
icmpInAddrMaskReps	Returns 1
icmpOutDestUnreachs	Returns 1
icmpOutTimeExcds	Returns 1
icmpOutParmProbs	Returns 1
icmpOutSrcQuenchs	Returns 1
icmpOutRedirects	Returns 1
icmpOutEchos	Returns 1
icmpOutEchoReps	Returns 1
icmpOutTimestamps	Returns 1
icmpOutTimestampReps	Returns 1
icmpOutAddrMasks	Returns 1
icmpOutAddrMaskReps	Returns 1
ifInUnknownProtos	Returns 0
ipAdEntBcastAddr	Returns 1
ipAdEntReasmMaxSiz	Returns 65535
ipRouteMetric1	Returns −1
ipRouteMetric2	Returns −1
ipRouteMetric3	Returns −1
ipRouteMetric4	Returns −1
ipRouteAge	Returns 0
ipRouteMetric5	Returns −1
ipNetToMediaType	Returns (3) dynamic
ipRoutingDiscards	Returns 0

The following variables return a fixed value of `0` for drivers not conforming to the GLD framework (see `gld(7D)`), including the old LAN drivers on SPARC machines:

`ifInOctets`	Returns 0
`ifInNUcastPkts`	Returns 0
`ifInDiscards`	Returns 0
`ifOutOctets`	Returns 0
`ifOutNUcastPkts`	Returns 0
`ifOutDiscards`	Returns 0

Schema Attributes

The following describes the attributes in the group and table definitions in the `/var/snmp/mib/sun.mib` file.

system

The `system` group reports statistics about a particular system (for example, a workstation or a printer).

`sysDescr` – A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. This value must only contain printable ASCII characters. (string[255])

`sysObjectID` – The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining what type of equipment is being managed. For example, if vendor “Flintstones, Inc.” was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its “Fred Router.” (objectid)

`sysUpTime` – Time (in hundredths of a second) since the network management portion of the system was last reinitialized. (timeticks)

`sysContact` – The textual identification of the contact person for this managed node, together with information on how to contact this person. (string[255])

`sysName` – An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name. (string[255])

`sysLocation` – The physical location of this node (for example, “telephone closet, 3rd floor” (string[255]))

`sysServices` – A value indicating the set of services that this entity primarily offers. (int) The value is a sum. This sum initially takes the value zero. Then, for each layer L in the range 1 through 7 for which this node performs transactions, 2 raised to (L - 1) is added to the sum. For example, a node that performs primarily routing functions would have a value of 4 (2*

*(3-1)). In contrast, a node that is a host offering application services would have a value of 72 (2**(4-1) + 2**(7-1)). Note that in the context of the Internet suite of protocols, values should be calculated accordingly:

Layer	Functionality
1	physical (such as repeaters)
2	datalink/subnetwork (such as bridges)
3	internet (such as IP gateways)
4	end-to-end (such as IP hosts)
7	applications (such as mail relays)

For systems including OSI protocols, Layers 5 and 6 may also be counted.

interfaces

The `interfaces` group reports the number of interfaces handled by the agent.

`ifNumber` – The number of network interfaces, regardless of their current state, present on this system. (int)

ifTable

The `ifTable` is a table of interface entries. The number of entries is given by the value of ifNumber.

`ifIndex` – A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization. (int)

`ifDescr` – A textual string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface. (string[255])

`ifType` – The type of interface, distinguished according to the physical/link protocol(s) immediately below the network layer in the protocol stack. (enum)

`ifMtu` – The size of the largest datagram that can be sent/received on the interface, specified in octets. For interfaces used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface. (int)

`ifSpeed` – An estimate of the interface's current bandwidth in bits-per-second. For interfaces that do not vary in bandwidth, or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. (gauge)

`if1hysAddress` – The interface's address at the protocol layer immediately below the network layer in the protocol stack. For interfaces without such an address (for example, a serial line), this object should contain an octet string of zero length. (octet[128])

`ifAdminStatus` – The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. (enum)

`if OperStatus` – The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. (enum)

`ifLastChange` – The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last reinitialization of the local network management subsystem, then this object contains a zero value. (timeticks)

`ifInOctets` – The total number of octets received on the interface, including framing characters. (counter) Returns a fixed value of 0.

`ifInUcastPkts` – The number of subnetwork-unicast packets delivered to a higher-layer protocol. (counter)

`ifInNUcastPkts` – The number of non-unicast (that is, subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol. (counter) Returns a fixed value of 0.

`ifInDiscards` – The number of inbound packets chosen to be discarded, even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. (counter) Returns a fixed value of 0.

`ifInErrors` – The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. (counter)

`ifInUnknownProtos` – The number of packets received via the interface that were discarded because of an unknown or unsupported protocol. (counter) Returns a fixed value of 0.

`ifOutOctets` – The total number of octets transmitted out of the interface, including framing characters. (counter) Returns a fixed value of 0.

`ifOutUcastPkts` – The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent. (counter)

`ifOutNUcastPkts` – The total number of packets that higher-level protocols requested be transmitted to a non- unicast (that is, a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent. (counter) Returns a fixed value of 0.

`ifOutDiscards` – The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. (counter) Returns a fixed value of 0.

`ifOutErrors` – The number of outbound packets that could not be transmitted because of errors. (counter)

`ifOutQLen` – The length of the output packet queue (in packets). (gauge)

`ifSpecific` – A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an Ethernet, then the value of this object refers to a document defining objects specific to Ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier. Any conformant implementation of ASN.1 and BER must be able to generate and recognize this value. (objectid)

atTable

`atTable` Address Translation tables contain the NetworkAddress to physical address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 has an algorithmic method). If all interfaces are of this type, then the Address Translation table is empty, that is, has zero entries.

`atIfIndex` – The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex. (int)

`atPhysAddress` – The media-dependent physical address. (octet[128]) Setting this object to a null string (one of zero length) has the effect of invaliding the corresponding entry in the atTable object. That is, it effectively dissociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant atPhysAddress object.

`atNetAddress` – The NetworkAddress (that is, the IP address) corresponding to the media-dependent physical address. (netaddress)

ip

The `ip` group reports statistics about the Internet Protocol (IP) group.

`ipForwarding` – The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. IP gateways forward datagrams. IP hosts do not— except those source-routed via the host. (enum)

Note that for some managed nodes, this object may take on only a subset of the values possible. Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to change this object to an inappropriate value.

`ipDefaultTTL` – The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity, whenever a TTL value is not supplied by the transport layer protocol. (int)

`ipInReceives` – The total number of input datagrams received from interfaces, including those received in error. (counter)

`ipInHdrErrors` – The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, and so on. (counter)

`ipInAddrErrors` – The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported Classes (for example, Class E). For entities that are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address. (counter)

`ipForwDatagrams` – The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities that do not act as IP Gateways, this counter will include only those packets that were Source-Routed via this entity, and the Source- Route option processing was successful. (counter)

`ipInUnknownProtos` – The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol. (counter)

`ipInDiscards` – The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded, for example, for lack of buffer space. Note that this counter does not include any datagrams discarded while awaiting reassembly. (counter)

`ipInDelivers` – The total number of input datagrams successfully delivered to IP user-protocols (including ICMP). (counter)

`ipOutRequests` – The total number of IP datagrams that local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams. (counter)

`ipOutDiscards` – The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (for example, for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion. (counter)

`ipOutNoRoutes` – The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this “no-route” criterion. Note that this includes any datagrams that a host cannot route because all its default gateways are down. (counter)

`ipReasmTimeout` – The maximum number of seconds that received fragments are held while they are awaiting reassembly at this entity. (int)

`ipReasmReqds` – The number of IP fragments received that needed to be reassembled at this entity. (counter)

`ipReasmOKs` – The number of IP datagrams successfully reassembled. (counter)

`ipReasmFails` – The number of failures detected by the IP reassembly algorithm, for whatever reason: timed out, errors, and the like. Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received. (counter)

`ipFragOKs` – The number of IP datagrams that have been successfully fragmented at this entity. (counter)

`ipFragFails` – The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, for example, because their “Don't Fragment” flag was set. (counter)

`ipFragCreates` – The number of IP datagram fragments that have been generated as a result of fragmentation at this entity. (counter)

`ipRoutingDiscards` – The number of routing entries that were chosen to be discarded even though they were valid. One possible reason for discarding such an entry could be to free-up buffer space for other routing entries. (counter) Returns a fixed value of 0.

ipAddrTable

`ipAddrTable` is a table of addressing information relevant to this entity's IP addresses.

`ipAdEntAddr` – The IP address to which this entry's addressing information pertains. (netaddress)

`ipAdEntIfIndex` – The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex. (int)

`ipAdEntNetMask` – The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1, and all the hosts bits set to 0. (netaddress)

`ipAdEntBcastAddr` – The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1. This value applies to both the subnet and network broadcasts addresses used by the entity on this (logical) interface. (int) Returns a fixed value of 1.

`ipAdEntReasmMaxSize` – The size of the largest IP datagram that this entity can reassemble from incoming IP fragmented datagrams received on this interface. (int) Returns a fixed value of 65535.

ipRouteTable

The `ipRouteTable` is this entity's IP Routing table.

`ipRouteDest` – The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple routes to a single destination can appear in the table, but

access to such multiple entries is dependent on the table- access mechanisms defined by the network management protocol in use. (netaddress)

`ipRouteIfIndex` – The index value that uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex. (int)

`ipRouteMetric1` – The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to –1. (int) Returns a fixed value of –1.

`ipRouteMetric2` – An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to –1. (int) Returns a fixed value of –1.

`ipRouteMetric3` – An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to –1. (int) Returns a fixed value of –1.

`ipRouteMetric4` – An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to –1. (int) Returns a fixed value of –1.

`ipRouteNextHop` – The IP address of the next hop of this route. (In the case of a route bound to an interface that is realized via a broadcast media, the value of this field is the agent's IP address on that interface.) (netaddress)

`ipRouteType` – The type of route. Note that the values direct (3) and indirect (4) refer to the notion of direct and indirect routing in the IP architecture. (enum)

Setting this object to the value invalid (2) has the effect of invalidating the corresponding entry in the ipRouteTable object. That is, it effectively dissociates the destination identified with said entry from the route identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipRouteType object.

`ipRouteProto` – The routing mechanism through which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols. (enum)

`ipRouteAge` – The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of “too old” can be implied except through knowledge of the routing protocol by which the route was learned. (int) Returns a fixed value of 0.

`ipRouteMask` – Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belongs to a class-A, B, or C network, and then using one of:

Mask	Network
255.0.0.0	class-A
255.255.0.0	class-B
255.255.255.0	class-C

If the value of the ipRouteDest is 0.0.0.0 (a default route), then the mask value is also 0.0.0.0. It should be noted that all IP routing subsystems implicitly use this mechanism. (netaddress)

`ipRouteMetric5` – An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to –1. (int) Returns a fixed value of –1.

`ipRouteInfo` – A reference to MIB definitions specific to the particular routing protocol responsible for this route, as determined by the value specified in the route's ipRouteProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER { 0 0 }, which is a syntactically valid object identifier. Any conformant implementation of ASN.1 and BER must be able to generate and recognize this value. (objectid)

ipNetToMediaTable

The `ipNetToMediaTable` is the IP Address Translation table used for mapping from IP addresses to physical addresses.

`ipNetToMediaIfIndex` – The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex. (int)

`ipNetToMediaPhysAddress` – The media-dependent physical address. (octet[128])

`ipNetToMediaNetAddress` – The IpAddress corresponding to the media- dependent physical address. (netaddress)

`ipNetToMediaType` – The type of mapping. (enum) Returns a fixed value of (3)dynamic. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively dissociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.

icmp

The `icmp` group reports statistics about the ICMP group.

`icmpInMsgs` – The total number of ICMP messages that the entity received. Note that this counter includes all those counted by icmpInErrors. (counter)

`icmpInErrors` – The number of ICMP messages that the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, and the like.). (counter)

`icmpInDestUnreachs` – The number of ICMP Destination Unreachable messages received. (counter)

`icmpInTimeExcds` – The number of ICMP Time Exceeded messages received. (counter)

`icmpInParmProbs` – The number of ICMP Parameter Problem messages received. (counter)

`icmpInSrcQuenchs` – The number of ICMP Source Quench messages received. (counter)

`icmpInRedirects` – The number of ICMP Redirect messages received. (counter)

`icmpInEchos` – The number of ICMP Echo (request) messages received. (counter)

`icmpInEchoReps` – The number of ICMP Echo Reply messages received. (counter)

`icmpInTimestamps` – The number of ICMP Timestamp (request) messages received. (counter)

`icmpInTimestampReps` – The number of ICMP Timestamp Reply messages received. (counter)

`icmpInAddrMasks` – The number of ICMP Address Mask Request messages received. (counter)

`icmpInAddrMaskReps` – The number of ICMP Address Mask Reply messages received. (counter)

`icmpOutMsgs` – The total number of ICMP messages that this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors. (counter)

`icmpOutErrors` – The number of ICMP messages that this entity did not send due to problems discovered within ICMP, such as a lack of buffers. This value should not include errors discovered outside the ICMP layer, such as the inability of IP to route the resultant datagram. In some implementations there may be no types of errors that contribute to this counter's value. (counter)

`icmpOutDestUnreachs` – The number of ICMP Destination Unreachable messages sent. (counter)

`icmpOutTimeExcds` – The number of ICMP Time Exceeded messages sent. (counter)

`icmpOutParmProbs` – The number of ICMP Parameter Problem messages sent. (counter)

icmpOutSrcQuenchs – The number of ICMP Source Quench messages sent. (counter)

icmpOutRedirects – The number of ICMP Redirect messages sent. For a host, this object will always be zero, since hosts do not send redirects. (counter)

icmpOutEchos – The number of ICMP Echo (request) messages sent. (counter)

icmpOutEchoReps – The number of ICMP Echo Reply messages sent. (counter)

icmpOutTimestamps – The number of ICMP Timestamp (request) messages sent. (counter)

icmpOutTimestampReps – The number of ICMP Timestamp Reply messages sent. (counter)

icmpOutAddrMasks – The number of ICMP Address Mask Request messages sent. (counter)

icmpOutAddrMaskReps – The number of ICMP Address Mask Reply messages sent. (counter)

tcp

The tcp group reports statistics about the TCP group.

tcpRtoAlgorithm – The algorithm used to determine the timeout value used for retransmitting unacknowledged octets. (enum)

tcpRtoMin – The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793. (int)

tcpRtoMax – The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the UBOUND quantity described in RFC 793. (int)

tcpMaxConn – The limit on the total number of TCP connections that the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value –1. (int)

tcpActiveOpens – The number of times that TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state. (counter)

tcpPassiveOpens – The number of times that TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state. (counter)

tcpAttemptFails – The number of times that TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state. (counter)

`tcpEstabResets` – The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state. (counter)

`tcpCurrEstab` – The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT. (gauge)

`tcpInSegs` – The total number of segments received, including those received in error. This count includes segments received on currently established connections. (counter)

`tcpOutSegs` – The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets. (counter)

`tcpRetransSegs` – The total number of segments retransmitted - that is, the number of TCP segments transmitted containing one or more previously transmitted octets. (counter)

`tcpInErrs` – The total number of segments received in error (for example, bad TCP checksums). (counter)

`tcpOutRsts` – The number of TCP segments sent containing the RST flag. (counter)

tcpConnTable

The `tcpConnTable` is a table containing TCP connection-specific information.

`tcpConnState` – The state of this TCP connection. (enum)

The only value that may be set by a management station is deleteTCB(12). Accordingly, it is appropriate for an agent to return a “badValue” response if a management station attempts to set this object to any other value.

If a management station sets this object to the value deleteTCB(12), then this has the effect of deleting the TCB (as defined in RFC 793) of the corresponding connection on the managed node. This results in immediate termination of the connection.

As an implementation-specific option, an RST segment may be sent from the managed node to the other TCP endpoint. (Note, however, that RST segments are not sent reliably.)

`tcpConnLocalAddress` – The local IP address for this TCP connection. For a connection in the listen state that is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used. (netaddress)

`tcpConnLocalPort` – The local port number for this TCP connection. (int)

`tcpConnRemAddress` – The remote IP address for this TCP connection. (netaddress)

`tcpConnRemPort` – The remote port number for this TCP connection. (int)

upd

The `udp` group reports statistics about the UDP group.

`udpInDatagrams` – The total number of UDP datagrams delivered to UDP users. (counter) Returns a fixed value of 0.

udpNoPorts – The total number of received UDP datagrams for which there was no application at the destination port. (counter) Returns a fixed value of 0.

udpInErrors – The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port. (counter)

udpOutDatagrams – The total number of UDP datagrams sent from this entity. (counter) Returns a fixed value of 0.

udpTable

The udpTable is a table containing UDP listener information.

udpLocalAddress – The local IP address for this UDP listener. For a UDP listener that is willing to accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used. (netaddress)

udpLocalPort – The local port number for this UDP listener. (int)

snmp

The snmp group reports statistics about the SNMP group.

snmpInPkts – The total number of Messages delivered to the SNMP entity from the transport service. (counter)

snmpOutPkts – The total number of SNMP Messages passed from the SNMP protocol entity to the transport service. (counter)

snmpInBadVersions – The total number of SNMP Messages delivered to the SNMP protocol entity that were for an unsupported SNMP version. (counter)

snmpInBadCommunityNames – The total number of SNMP Messages delivered to the SNMP protocol entity that used a SNMP community name not known to said entity. (counter)

snmpInBadCommunityUses – The total number of SNMP Messages delivered to the SNMP protocol entity, which represented an SNMP operation not allowed by the SNMP community named in the Message. (counter)

snmpInASNParseErrs – The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP Messages. (counter)

snmpInTooBigs – The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “tooBig.” (counter)

snmpInNoSuchNames – The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “noSuchName.” (counter)

snmpInBadValues – The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is “badValue.” (counter)

`snmpInReadOnlys` – The total number valid SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is "readOnly." It should be noted that it is a protocol error to generate an SNMP PDU that contains the value "readOnly" in the error-status field. This object is provided as a means of detecting incorrect implementations of the SNMP. (counter)

`snmpInGenErrs` – The total number of SNMP PDUs delivered to the SNMP protocol entity for which the value of the error-status field is "genErr." (counter)

`snmpInTotalReqVars` – The total number of MIB objects successfully retrieved by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs. (counter)

`snmpInTotalSetVars` – The total number of MIB objects successfully altered by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs. (counter)

`snmpInGetRequests` – The total number of SNMP Get-Request PDUs accepted and processed by the SNMP protocol entity. (counter)

`snmpInGetNexts` – The total number of SNMP Get-Next PDUs accepted and processed by the SNMP protocol entity. (counter)

`snmpInSetRequests` – The total number of SNMP Set-Request PDUs accepted and processed by the SNMP protocol entity. (counter)

`snmpInGetResponses` – The total number of SNMP Get-Response PDUs accepted and processed by the SNMP protocol entity. (counter)

`snmpInTraps` – The total number of SNMP Trap PDUs accepted and processed by the SNMP protocol entity. (counter)

`snmpOutTooBigs` – The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status field is "tooBig." (counter)

`snmpOutNoSuchNames` – The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status is "noSuchName." (counter)

`snmpOutBadValues` – The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status field is "badValue." (counter)

`snmpOutGenErrs` – The total number of SNMP PDUs generated by the SNMP protocol entity for which the value of the error-status field is "genErr." (counter)

`snmpOutGetRequests` – The total number of SNMP Get-Request PDUs which have been generated by the SNMP protocol entity. (counter)

`snmpOutGetNexts` – The total number of SNMP Get-Next PDUs generated by the SNMP protocol entity. (counter)

snmpOutSetRequests – The total number of SNMP Set-Request PDUs generated by the SNMP protocol entity. (counter)

snmpOutGetResponses – The total number of SNMP Get-Response PDUs generated by the SNMP protocol entity. (counter)

snmpOutTraps – The total number of SNMP Trap PDUs generated by the SNMP protocol entity. (counter)

snmpEnableAuthenTraps – Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information. As such, it provides a means whereby all authentication-failure traps may be disabled. (enum)

Note that this object must be stored in non-volatile memory, so that it remains constant between reinitializations of the network management system.

The following are Sun-specific group and table definitions.

sunSystem

The sunSystem group reports general system information.

agentDescr – The SNMP agent's description of itself. (string[255])

hostID – The unique Sun hardware identifier. The value returned is four byte binary string. (octet[4])

motd – The first line of /etc/motd. (string[255])

unixTime – The UNIX system time. Measured in seconds since January 1, 1970 UTC. (counter)

sunProcessTable

The sunProcessTable table reports UNIX process table information.

psProcessID – The process identifier for this process. (int)

psParentProcessID – The process identifier of this process's parent. (int)

psProcessSize – The combined size of the data and stack segments (in kilobytes.) (int)

psProcessCpuTime – The CPU time (including both user and system time) consumed so far. (int)

psProcessState – The run-state of the process. (octet[4])

R	Runnable
T	Stopped
P	In page wait

D	Non-interruptable wait
S	Sleeping (less than 20 seconds)
I	Idle (more than 20 seconds)
Z	Zombie

`psProcessWaitChannel` – Reason process is waiting. (octet[16])

`psProcessTTY` – Terminal, if any, controlling this process. (octet[16])

`psProcessUserName` – Name of the user associated with this process. (octet[16])

`psProcessUserID` – Numeric form of the name of the user associated with this process. (int)

`psProcessName` – Command name used to invoke this process. (octet[64])

`psProcessStatus` – Setting this variable will cause a signal of the set value to be sent to the process. (int)

sunHostPerf

The `sunHostPerf` group reports hostperf information.

`rsUserProcessTime` – Total number of timeticks used by user processes since the last system boot. (counter)

`rsNiceModeTime` – Total number of timeticks used by “nice” mode since the last system boot. (counter)

`rsSystemProcessTime` – Total number of timeticks used by system processes since the last system boot. (counter)

`rsIdleModeTime` – Total number of timeticks in idle mode since the last system boot. (counter)

`rsDiskXfer1` – Total number of disk transfers since the last boot for the first of four configured disks. (counter)

`rsDiskXfer2` – Total number of disk transfers since the last boot for the second of four configured disks. (counter)

`rsDiskXfer3` – Total number of disk transfers since the last boot for the third of four configured disks. (counter)

`rsDiskXfer4` – Total number of disk transfers since the last boot for the fourth of four configured disks. (counter)

`rsVPagesIn` – Number of pages read in from disk. (counter)

`rsVPagesOut` – Number of pages written to disk. (counter)

`rsVSwapIn` – Number of pages swapped in. (counter)

`rsVSwapOut` – Number of pages swapped out. (counter)

`rsVIntr` – Number of device interrupts. (counter)

`rsIfInPackets` – Number of input packets. (counter)

`rsIfOutPackets` – Number of output packets. (counter)

`rsIfInErrors` – Number of input errors. (counter)

`rsIfOutErrors` – Number of output errors. (counter)

`rsIfCollisions` – Number of output collisions. (counter)

ファイル

`/etc/snmp/conf/snmpd.conf`	configuration information
`/etc/snmp/conf/mibiisa.acl`	access control file
`/var/snmp/mib/sun.mib`	standard SNMP MIBII file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/snmp/mibiisa
Interface Stability	Obsolete

関連項目

`inetd(1M)`, `select(3C)`, `recvfrom(3SOCKET)`, `sendto(3SOCKET)`, `attributes(5)`, `gld(7D)`

診断

`cannot dispatch request`	The proxy cannot dispatch the request. The rest of the message indicates the cause of the failure.
`select(3C) failed`	A `select(3C)` call failed. The rest of the message indicates the cause of the failure.
`sendto(3SOCKET) failed`	A `sendto(3SOCKET)` call failed. The rest of the message indicates the cause of the failure.
`recvfrom(3SOCKET) failed`	A `recvfrom(3SOCKET)` call failed. The rest of the message indicates the cause of the failure.
`no response from system`	The SNMP agent on the target system does not respond to SNMP requests. This error might indicate that the

	SNMP agent is not running on the target system, the target system is down, or the network containing the target system is unreachable.
`response too big`	The agent could not fit the results of an operation into a single SNMP message. Split large groups or tables into smaller entities.
`missing attribute`	An attribute is missing from the requested group.
`bad attribute type`	An object attribute type received from the SNMP agent that does not match the attribute type specified by the proxy agent schema. The rest of the message indicates the expected type and received type.
`cannot get sysUpTime`	The proxy agent cannot get the variable *sysUpTime* from the SNMP agent.
`sysUpTime type bad`	The variable *sysUpTime* received from the SNMP agent has the wrong data type.
`unknown SNMP error`	An unknown SNMP error was received.
`bad variable value`	The requested specified an incorrect syntax or value for a set operation.
`variable is read only`	The SNMP agent did not perform the set request because a variable to set may not be written.
`general error`	A general error was received.
`cannot make request PDU`	An error occurred building a request PDU.
`cannot make request varbind list`	An error occurred building a request variable binding list.
`cannot parse response PDU`	An error occurred parsing a response PDU.

`request ID - response ID mismatch`	The response ID does not match the request ID.
`string contains non-displayable characters`	A displayable string contains non-displayable characters.
`cannot open schema file`	An error occurred opening the proxy agent schema file.
`cannot parse schema file`	The proxy agent couldn't parse the proxy agent schema file.
`cannot open host file`	An error occurred opening the file associated with the *na.snmp.hostfile* keyword in `/etc/snmp/conf/snmpd.conf`
`cannot parse host file`	The proxy agent was unable to parse the file associated with the *na.snmp.hostfile* keyword in `/etc/snmp/conf/snm.conf`.
`attribute unavailable for set operations`	The set could not be completed because the attribute was not available for set operations.

使用上の留意点

The `mibiisa` utility returns the wrong interface speed for the SBUS FDDI interface (for example, “bf0”).

The `mibiisa` utility does not return a MAC address for the SBUS FDDI interface (for example, “bf0”).

Process names retrieved from `mibiisa` contain a leading blank space.

When you change attribute values in the system group with an SNMP set request, the change is effective only as long as `mibiisa` is running. `mibiisa` does not save the changes to `/etc/snmp/conf/snmpd.conf`.

名前

mkbootmedia – create bootable Solaris ISO image

形式

`/usr/bin/mkbootmedia -v [-l label] media-root iso`

機能説明

The `mkbootmedia` utility takes *media-root* (the root of an on-disk Solaris install media) as input and creates a bootable Solaris ISO image in the file *iso*, using `mkisofs(8)`. The file can then be burned onto a CD/DVD with utilities such as `cdrw(1)` or `cdrecord(1)`. (Neither `mkisofs(8)` nor `cdrecord(1)` are SunOS man pages.)

注意 – The directory tree *media-root* must contain the file `boot/grub/stage2_eltorito`, which will be written to the media boot sectors. This file will be modified with some boot information, thus it must be writable. If necessary, first save a copy prior to running this utility.

オプション

The following options are supported:

`-l` *label*
: Sets *label* as the label/volume name of the ISO image.

`-v`
: Verbose. Multiple `-v` options increase verbosity.

オペランド

The following operands are supported:

media-root
: Top-level directory of an on-disk Solaris install media.

iso
: Name of the output file which will contain the resulting ISO image.

使用例

例 1 Creating an ISO Image and Burning a CD/DVD

The following commands create an ISO image from the content of `s10u1` and burn the image to a CD/DVD.

```
# /usr/bin/mkbootmedia s10u1 s10u1.iso
# /usr/bin/cdrw -i s10u1.iso
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`cdrw(1)`, `attributes(5)`

`mkisofs(8)`, (`/usr/share/man/man8/mkisofs.8`), in the `SUNWfsman` package (not a SunOS man page)

名前

mkdevalloc – Make device_allocate entries

形式

```
/usr/sbin/mkdevalloc
```

機能説明

The `mkdevalloc` command writes to standard out a set of `device_allocate(4)` entries describing the system's frame buffer, audio, and removable media devices.

The `mkdevalloc` command is used by the device allocation service:

```
svc:/system/device/allocate:default
```

...to create or update the `device_allocate(4)` file. The device allocation service is managed by `smf(5)` and described in `device_allocate(1M)`.

Entries are generated based on the device special files found in `/dev`. For the different categories of devices, the `mkdevalloc` command checks for the following files under `/dev`:

```
audio           /dev/audio, /dev/audioctl, /dev/sound/...
tape            /dev/rst*, /dev/nrst*, /dev/rmt/...
removable disk  /dev/sr*, /dev/nsr*, /dev/dsk/c0t?d0s?, /dev/rdsk/c0t?d0s?
frame buffer    /dev/fb
```

All entries set the *device-minimum* and *device-maximum* fields to the hex representations of `ADMIN_LOW` and `ADMIN_HIGH`, respectively. The *device-authorization* field is set to `solaris.device.allocate`, except for the `framebuffer` entry, where it is set to `*`. The *device-name*, *device-type* and *device-clean* fields are set to the following values:

```
                device-name      device-type   device-clean

audio           audio            audio         audio_clean_wrapper
tape            mag_tape_0,1,... st            st_clean
floppy          floppy_0,1,...   fd            disk_clean
removable disk  cdrom_0,1,...    sr            disk_clean
frame buffer    framebuffer      fb            /bin/true
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Obsolete

関連項目

`allocate(1)`, `device_allocate(1M)`, `device_allocate(4)`, `attributes(5)`, `smf(5)`

注意事項

`mkdevalloc` might not be supported in a future release of the Solaris operating system.

名前

mkdevmaps – make device_maps entries

形式

```
/usr/sbin/mkdevmaps
```

機能説明

The mkdevmaps command writes to standard out a set of device_maps(4) entries describing the system's frame buffer, audio, and removable media devices.

The mkdevmaps command is used by the device allocation service:

```
svc:/system/device/allocate:default
```

...to create or update the device_maps(4) file. The device allocation service is managed by smf(5) and described in device_allocate(1M).

Entries are generated based on the device special files found in /dev. For the different categories of devices, the mkdevmaps command checks for the following files under /dev:

```
audio           /dev/audio, /dev/audioctl, /dev/sound/...
tape            /dev/rst*, /dev/nrst*, /dev/rmt/...
removable disk  /dev/dsk/c0t?d0s?, /dev/rdsk/c0t?d0s?
frame buffer    /dev/fb
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Obsolete

関連項目

allocate(1), device_allocate(1M), device_maps(4), attributes(5), smf(5)

注意事項

mkdevmaps might not be supported in a future release of the Solaris operating system.

名前

mkfifo – make FIFO special file

形式

`/usr/bin/mkfifo [-m mode] path...`

機能説明

The `mkfifo` utility creates the FIFO special files named by its argument list. The arguments are taken sequentially, in the order specified; and each FIFO special file is either created completely or, in the case of an error or signal, not created at all.

If errors are encountered in creating one of the special files, `mkfifo` writes a diagnostic message to the standard error and continues with the remaining arguments, if any.

The `mkfifo` utility calls the library routine `mkfifo(3C)`, with the *path* argument is passed as the *path* argument from the command line, and *mode* is set to the equivalent of `a=rw`, modified by the current value of the file mode creation mask `umask(1)`.

オプション

The following option is supported:

`-m` *mode* Set the file permission bits of the newly-created FIFO to the specified *mode* value. The *mode* option-argument will be the same as the *mode* operand defined for the `chmod(1)` command. In <*symbolic*mode> strings, the *op* characters + and – will be interpreted relative to an assumed initial mode of `a=rw`.

オペランド

The following operand is supported:

file A path name of the FIFO special file to be created.

使用法

See `largefile(5)` for the description of the behavior of `mkfifo` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

環境変数

See `environ(5)` for descriptions of the following environment variables that affect the execution of `mkfifo`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

終了ステータス

The following exit values are returned:

`0` All the specified FIFO special files were created successfully.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See `standards(5)`.

関連項目 chmod(1), umask(1), mkfifo(3C), attributes(5), environ(5), largefile(5), standards(5)

名前

mkfile – ファイルの作成

形式

`mkfile [-nv] size [t | g | k | b | m] filename...`

機能説明

`mkfile` は、NFS マウント上のファイル、あるいはローカルファイルをスワップ領域として使用するのに適した 1 つまたは複数のファイルを作成します。root ユーザーとして `mkfile()` を実行すると、デフォルトでスティッキービットが設定され、ファイルはゼロでパディングされます。root 以外のユーザーは、`mkfile()` を実行するときに、`chmod(1)` を使用して、手動でスティッキービットを設定する必要があります。デフォルトでは `size` の単位はバイトですが、`t`、`g`、`k`、`b`、または `m` のサフィックスを使用すると、テラバイト、ギガバイト、キロバイト、ブロック、またはメガバイト単位でも指定できます。サフィックスには大文字または小文字を使用できます。

オプション

`-n` 空の *filename* を作成します。サイズは出力されますが、ディスクブロックはデータが書き込まれるまで割り当てられません。このオプションで作成したファイルは、ローカルの UFS マウント上でスワップできません。

`-v` 冗長。作成したファイルの名前とサイズを報告します。

終了ステータス

次の終了値が返されます。

`0` 成功。

`>0` エラーが発生。

使用法

2G バイト (2^{31} バイト) 以上のファイルを検出した場合の `mkfile` の動作については、`largefile(5)` を参照してください。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

`chmod(1)`, `swap(1M)`, `attributes(5)`, `largefile(5)`

名前

mkfs – construct a file system

形式

```
mkfs [-F FSType] [generic_options]
     [-o FSType-specific_options] raw_device_file
     [operands]
```

機能説明

The `mkfs` utility constructs a file system on the *raw_device_file* by calling the specific `mkfs` module indicated by `-F` *FSType*.

`Note`: ufs file systems are normally created with the `newfs(1M)` command.

generic_options are independent of file system type. *FSType-specific_options* is a comma-separated list of *keyword=value* pairs (with no intervening spaces), which are *FSType*-specific. *raw_device_file* specifies the disk partition on which to write the file system. It is required and must be the first argument following the *specific_options* (if any). *operands* are *FSType*-specific. See the *FSType*-specific manual page of `mkfs` (for example, `mkfs_ufs(1M)`) for a detailed description.

オプション

The following are the generic options for `mkfs`:

`-F` Specify the *FSType* to be constructed. If `-F` is not specified, the *FSType* is determined from `/etc/vfstab` by matching the *raw_device_file* with a `vfstab` entry, or by consulting the `/etc/default/fs` file.

`-V` Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided and adding to them information derived from `/etc/vfstab` or `/etc/default/fs`. This option may be used to verify and validate the command line.

`-m` Return the command line which was used to create the file system. The file system must already exist. This option provides a means of determining the command used in constructing the file system.

`-o` Specify *FSType*-specific options. See the manual page for the `mkfs` module specific to the file system type.

使用法

See `largefile(5)` for the description of the behavior of `mkfs` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/default/fs` Default file system type. Default values can be set for the following flags in `/etc/default/fs`. For example: `LOCAL=ufs`

- `LOCAL` The default partition for a command if no *FSType* is specified.

`/etc/vfstab` List of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

mkfs_ufs(1M), newfs(1M), vfstab(4), attributes(5), largefile(5)

Manual pages for the FSType-specific modules of mkfs.

注意事項

This command might not be supported for all *FSTypes*.

You can use lofiadm to create a file that appears to a mkfs command as a raw device. You can then use a mkfs command to create a file system on that device. See lofiadm(1M) for examples of creating a UFS and a PC (FAT) file system (using mkfs_ufs(1M) and mkfs_pcfs(1M)) on a device created by lofiadm.

名前

mkfs_pcfs – construct a FAT file system

形式

```
mkfs -F pcfs [generic_options] [-o FSType_specific_options]
      raw_device_file
```

機能説明

The `pcfs`-specific module of `mkfs` constructs a File Allocation Table (FAT) on removable media (JAZ disk, ZIP disk, PCMCIA card), a hard disk, or a file (see NOTES). FATs are the standard MS-DOS and Windows file system format.

`mkfs` for `pcfs` determines an appropriate FAT size for the medium, then it installs an initial boot sector and an empty FAT. A sector size of 512 bytes is used. `mkfs` for `pcfs` can also install the initial file in the file system (see the `pcfs`-specific `-o i` option). This first file can optionally be marked as read-only, system, and/or hidden.

If you want to construct a FAT with `mkfs` for `pcfs` on a medium that is not formatted, you must first perform a low-level format on the medium with `format(1M)`. The media must also be partitioned with the `fdisk(1M)` utility. Note that all existing data on a disk partition, if any, is destroyed when a new FAT is constructed.

generic_options are supported by the generic `mkfs` command. See `mkfs(1M)` for a description of these options.

raw_device_file indicates the device on which to write unless the `-o N` option has been specified, or if the `-V` or `-m` generic options are passed from the generic `mkfs` module.

オプション

See `mkfs(1M)` for the list of supported generic options.

The following options are supported:

`-o` *FSType_specific_options*
: Specify `pcfs` file system-specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored.

b=*label*
: Label the media with volume label. The volume label is restricted to 11 uppercase characters.

B=*filename*
: Install *filename* as the boot loader in the file system's boot sector. If you don't specify a boot loader, an MS-DOS boot loader is installed. The MS-DOS boot loader requires specific MS-DOS system files to make a disk bootable. See `NOTES` for more information.

fat=*n*
: The size of a FAT entry. Currently, 12, 16, and 32 are valid values. The default is 16.

h
: Mark the first file installed as a hidden file. The `-i` option must also be specified.

hidden=*n*
: Set the number of hidden sectors to *n*. This is the number of sectors on the physical disk preceding the start of the volume (which is the boot sector itself). This defaults to a computed valued (based on the fdisk table) for disks. This option may be used only in conjunction with the `nofdisk` option.

i=*filename*
: Install *filename* as the initial file in the new file system. The initial file's contents are guaranteed to occupy consecutive clusters at the start of the files area. When creating bootable media, a boot program should be specified as the initial file.

nofdisk
: Do not attempt to find an `fdisk` table on the medium. Instead rely on the `size` option for determining the partition size. By default, the created FAT is 16 bits and begins at the first sector of the device. This origination sector can be modified with the hidden option (`-h`).

nsect=*n*
: The number of sectors per track on the disk. If not specified, the value is determined by using a `dkio(7I)` ioctl to get the disk geometry.

ntrack=*n*
: The number of tracks per cylinder on the disk. If not specified, the value is determined by using a `dkio(7I)` ioctl to get the disk geometry.

N
: No execution mode. Print normal output, but do not actually write the file system to the medium. This is most useful when used in conjunction with the verbose option.

r
: Mark the first file installed as read-only. The `-i` option must also be specified.

reserve=*n*
: Set the number of reserved sectors to *n*. This is the number of sectors in the volume, preceding the start of the first FAT, including the boot sector. The value should always be at least 1, and the default value is exactly 1.

s
: Mark the first file installed as a system file. The `-i` option must also be specified.

size=*n*
: The number of sectors in the file system. If not specified, the value is determined from the size of the partition given in the fdisk table.

spc=*n*
: The size of the allocation unit for space within the file system, expressed as a number of sectors. The default value depends on the FAT entry size and the size of the file system.

v
: Verbose output. Describe, in detail, operations being performed.

ファイル

raw_device_file

The device on which to build the FAT.

`mkfs_pcfs` supports MBR (Master Boot Record) partitions and GPT (GUID Partition Table) partitions. GPT is part of the EFI (Extensible Firmware Interface) standard. For both x86 and SPARC, for a GPT-labeled disk, you can specify the partition using the logical device pathname with no suffix, for example, `/dev/rdsk/c0t0d0s0`. In GPT, this corresponds to the first partition on the disk.

On x86 for MBR partitions, you can specify the proper partition using the logical device pathname corresponding to the partition. For example, `/dev/rdsk/c0t0d0p1` corresponds to first partition in the MBR, or `/dev/rdsk/c0t0d0p5` corresponds to first logical partition in the extended partition. Alternatively, using a suffix is also acceptable. For example, in `/dev/rdsk/c0t0d0p0:c`, `mkfs_pcfs` recognizes `:c` as the first partition that can accept a FAT file system.

For removable media with MBR partitions on SPARC, you need to specify a disk device name with a suffix to indicate the proper partition. For example, in the name `/dev/rdsk/c0t0d0s2:c`, the `:c` suffix indicates that the partition can accept the new FAT.

For a file, *raw_device_file* is the block device name returned by `lofiadm(1M)`.

使用例

The media in these examples must be formatted before running `mkfs` for `pcfs`. See DESCRIPTION for more details.

例 1 Creating a FAT File System on a Disk

The following command creates a FAT file system on the second fdisk partition of a disk attached to an x86 based system:

```
mkfs -F pcfs /dev/rdsk/c0d0p0:d
```

例 2 Creating a FAT File System on a ZIP Disk

The following command creates a FAT file system on a ZIP disk located on a SPARC based system:

```
mkfs -F pcfs /dev/rdsk/c0t4d0s2:c
```

例 3 Creating a FAT File System on a JAZ Disk

The following command creates a FAT file system on a JAZ disk located on a SPARC based system and overrides the sectors/track and tracks/cylinder values obtained from the device's controller:

```
mkfs -F pcfs -o nsect=32,ntrack=64 /dev/rdsk/c0t3d0s2:c
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/pcfs
Interface Stability	Committed

関連項目

fdisk(1M), format(1M), lofiadm(1M), mkfs(1M), attributes(5), dkio(7I)

注意事項

You can use lofiadm to create a file that appears to a mkfs command (for example, mkfs_pcfs or mkfs_ufs) as a raw device. You can then use a mkfs command to create a file system on that device. See lofiadm(1M) for examples of creating a UFS and a PC (FAT) file system on a device created by lofiadm.

名前

mkfs_udfs – construct a udfs file system

形式

```
mkfs -F udfs [generic_options] [-o specific_options] raw_device_file
      [size]
```

機能説明

This is the universal disk format file system (udfs) -specific module of the `mkfs` command. `mkfs` constructs a `udfs` file system with a root directory.

オプション

See `mkfs`(1M) for the list of supported *generic_options*.

The following options are supported:

`-o` *specific_options*

Specify a `udfs`-specific option. Specify `udfs` file system specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored.

The following *specific_options* are available:

N

Print the file system parameters without actually creating the file system.

label=*string*

Specify the label to be written into the volume header structures. Specify *string* as the name of the label. If *string* is not specified, a default *string* is generated in the form of `*NoLabel*`.

オペランド

The following operands are supported:

raw_device_file

Specify the disk partition on which to write.

size

Specify the number of 512-byte blocks in the file system.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/udfs

関連項目

`fsck(1M)`,`mkfs(1M)`, `attributes(5)`

診断

`not currently a valid file system`

The specified device does not contain a valid `udfs` file system.

`Invalid size: larger than the partition size`

Number of blocks given as parameter to create the file system is larger than the size of the device specified.

`is mounted can’t mkfs`

Device is in use, cannot create file system when the device is in use.

```
preposterous size
```

Negative size parameter provided is invalid.

```
sector size must be between 512, 8192 bytes
```

Sector size given is not in the valid range.

```
Volume integrity sequence descriptors too long
File set descriptor too long.
```

Not enough space to create volume integrity sequence or file set descriptor.

```
mkfs: argument out of range
```

One of the arguments is out of range.

```
mkfs: bad numeric arg
```

One of the arguments is potentially a bad numeric.

注意事項

You can use `lofiadm` to create a file that appears to a `mkfs` command (for example, `mkfs_pcfs` or `mkfs_ufs`) as a raw device. You can then use a `mkfs` command to create a file system on that device. See `lofiadm(1M)` for examples of creating a UFS and a PC (FAT) file system on a device created by `lofiadm`.

名前

mkfs_ufs – construct a UFS file system

形式

`mkfs -F ufs [`*generic_options*`] [-o` *FSType_specific_options*`]` *raw_device_file* `[`*size*`]`

機能説明

The UFS-specific module of `mkfs` builds a UFS file system with a root directory and a `lost+found` directory (see `fsck(1M)`).

The UFS-specific `mkfs` is rarely run directly. Use the `newfs(1M)` command instead.

raw_device_file indicates the disk partition on which to create the new file system. If the `-o N`, `-V`, or `-m` options are specified, the *raw_device_file* is not actually modified. *size* specifies the number of disk sectors in the file system, where a disk sector is usually 512 bytes. This argument must follow the *raw_device_file* argument and is required (even with `-o N`), unless the `-V` or `-m` generic options are specified.

generic_options are supported by the generic `mkfs` command. See `mkfs(1M)` for a description of these options.

オプション

The following generic options are supported:

`-m` Print the command line that was used to create the existing file system.

`-V` Print the current `mkfs` command line.

オプション

The following UFS-specific options are supported:

`-o` Use one or more of the following values separated by commas (with no intervening spaces) to specify UFS-specific options:

`apc=`*n* The number of alternate sectors per cylinder to reserve for bad block replacement for SCSI devices only. The default is `0`.

This option is not applicable for disks with EFI labels and is ignored.

`bsize=`*n* The logical block size of the file system in bytes, either `4096` or `8192`. The default is `8192`. The sun4u architecture does not support the `4096` block size.

`calcbinsb` Sends to stdout a binary (machine-readable) version of the superblock that would be used to create a file system with the specified configuration parameters.

`calcsb` Sends to stdout a human-readable version of the superblock that would be used to create a file system with the specified configuration parameters.

`cgsize=`*n* The number of cylinders per cylinder group, ranging from `16` to `256`. The default is calculated by dividing the number of sectors in the file system by the number of sectors in a gigabyte. Then, the result is multiplied by `32`. The default value is always between `16` and `256`.

The per-cylinder-group meta data must fit in a space no larger than what is available in one logical file system block. If too large a `cgsize` is requested, it is changed by the minimum amount necessary.

`fragsize=`*n*

The smallest amount of disk space in bytes that can be allocated to a file. *fragsize* must be a power of 2 divisor of *bsize*, where:

bsize / *fragsize* is 1, 2, 4, or 8.

This means that if the logical block size is `4096`, legal values for *fragsize* are `512`, `1024`, `2048`, and `4096`. When the logical block size is `8192`, legal values are `1024`, `2048`, `4096`, and `8192`. The default value is `1024`.

For file systems greater than 1 terabyte or for file systems created with the `mtb=y` option, *fragsize* is forced to match block size (*bsize*).

`free=`*n*

The minimum percentage of free space to maintain in the file system between 0% and 99%, inclusively. This space is off-limits to users. Once the file system is filled to this threshold, only the superuser can continue writing to the file system.

The default is ((64 Mbytes/partition size) * 100), rounded down to the nearest integer and limited between 1% and 10%, inclusively.

This parameter can be subsequently changed using the `tunefs(1M)` command.

`gap=`*n*

Rotational delay. This option is obsolete in the Solaris 10 release. The value is always set to `0`, regardless of the input value.

`maxcontig=`*n*

The maximum number of logical blocks, belonging to one file, that are allocated contiguously. The default is calculated as follows:

`maxcontig =` *disk drive maximum transfer size / disk block size*

If the disk drive's maximum transfer size cannot be determined, the default value for `maxcontig` is calculated from kernel parameters as follows:

If `maxphys` is less than `ufs_maxmaxphys`, which is typically 1 Mbyte, then `maxcontig` is set to `maxphys`. Otherwise, `maxcontig` is set to `ufs_maxmaxphys`.

You can set `maxcontig` to any positive integer value.

The actual value will be the lesser of what has been specified and what the hardware supports.

You can subsequently change this parameter by using tunefs(1M).

`mtb=y`

Set the parameters of the file system to allow eventual growth to over a terabyte in total file system size. This option sets *fragsize* to be the same as *bsize*, and sets *nbpi* to 1 Mbyte, unless the `-i` option is used to make it even larger. If you explicitly set the *fragsize* or *nbpi* parameters to values that are incompatible with this option, the user-supplied value of *fragsize* or *nbpi* is ignored.

`N`

Print out the file system parameters that would be used to create the file system without actually creating the file system.

`nbpi=`*n*

The number of bytes per inode, which specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the number of inodes to create.

This value should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used. To create more inodes, a smaller number should be given. The default is `2048`.

The number of inodes can increase if the file system is expanded with the `growfs` command.

`nrpos=`*n*

The number of different rotational positions in which to divide a cylinder group. The default is `8`.

This option is not applicable for disks with EFI labels and is ignored.

`nsect=`*n*

The number of sectors per track on the disk. The default is `32`.

`ntrack=`*n*

The number of tracks per cylinder on the disk. The default is `16`.

This option is not applicable for disks with EFI labels and is ignored.

`opt=`*s* | *t*

The file system can either be instructed to try to minimize the *time* spent allocating blocks, or to try to minimize the *space* fragmentation on the disk. The default is *time*.

This parameter can be subsequently changed with the tunefs(1M) command.

`rps=`*n*

The rotational speed of the disk, in revolutions per second. The default is `60`.

Note that you specify *rps* for `mkfs` and *rpm* for `newfs`.

This option is not applicable for disks with EFI labels and is ignored.

Alternatively, parameters can be entered as a list of space-separated values (without keywords) whose meaning is positional. In this case, the `-o` option is omitted and the list follows the size operand. This is the way `newfs` passes the parameters to `mkfs`.

オペランド

The following operands are supported:

raw_device_file The disk partition on which to write.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fsck(1M)`, `mkfs(1M)`, `newfs(1M)`, `tunefs(1M)`, `dir_ufs(4)`, `attributes(5)`, `ufs(7FS)`

診断

The following error message typically occurs with very high density disks. On such disks, the file system structure cannot encode the proper disk layout information. However, such disks have enough onboard intelligence to make up for any layout deficiencies, so there is no actual impact on performance. The warning that performance might be impaired can be safely ignored.

```
Warning: insufficient space in super block for
rotational layout tables with nsect sblock.fs_nsect
and ntrak sblock.fs_ntrak. (File system performance may be impaired.)
```

The following error message occurs when the disk geometry results in a situation where the last truncated cylinder group cannot contain the correct number of data blocks. Some disk space is wasted.

```
Warning: inode blocks/cyl group (grp) >= data blocks (num) in last cylinder
```

If there is only one cylinder group and if the above condition holds true, `mkfs` fails with the following error:

```
File system creation failed. There is only one cylinder group and that is
not even big enough to hold the inodes.
```

The following error message occurs when the best calculated file system layout is unable to include the last few sectors in the last cylinder group. This is due to the interaction between how much space is used for various pieces of meta data and the total blocks available in a cylinder group. Modifying `nbpi` and `cpg` might reduce this number, but it is rarely worth the effort.

```
Warning: num sector(s) in last cylinder group unallocated
```

注意事項

You can use `lofiadm` to create a file that appears to the `mkfs` command (for example, `mkfs_pcfs` or `mkfs_ufs`) as a raw device. You can then use the `mkfs` command to create a file system on that device. See `lofiadm(1M)` for examples of creating a UFS and a PC (FAT) file system on a device created by `lofiadm`.

Both the block and character devices, such as devices in `/dev/dsk` and `/dev/rdsk`, must be available prior to running the `mkfs` command.

名前

mknod – make a special file

形式

`mknod` *name* `b` *major* *minor*

`mknod` *name* `c` *major* *minor*

`mknod` *name* `p`

機能説明

`mknod` makes a directory entry for a special file.

オプション

The following options are supported:

`b` Create a block-type special file.

`c` Create a character-type special file.

`p` Create a FIFO (named pipe).

オペランド

The following operands are supported:

major The *major* device number.

minor The *minor* device number; can be either decimal or octal. The assignment of major device numbers is specific to each system. You must be the super-user to use this form of the command.

name A special file to be created.

使用法

See `largefile(5)` for the description of the behavior of `mknod` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`ftp(1)`, `mknod(2)`, `symlink(2)`, `attributes(5)`, `largefile(5)`

注意事項

If `mknod(2)` is used to create a device, the major and minor device numbers are always interpreted by the kernel running on that machine.

With the advent of physical device naming, it would be preferable to create a symbolic link to the physical name of the device (in the `/devices` subtree) rather than using `mknod`.

名前

mkntfs – NTFS ファイルシステムの作成

形式

```
mkntfs [options] device [number_of_sectors]

mkntfs [-C] [-c cluster-size] [-F] [-f] [-H heads] [-h] [-I]
     [-L volume-label] [-l] [-n] [-p part-start-sect] [-Q] [-q]
     [-S sectors-per-track] [-s sector-size] [-T] [-V] [-v]
     [-z mft-zone-multiplier] [--debug] device [number-of-sectors]
```

機能説明

`mkntfs` ユーティリティーは、デバイス (通常はディスクパーティション)、またはファイルに NTFS ファイルシステムを作成するのに使用します。*device* オペランドは、デバイスに対応する特殊ファイルです (例: `/dev/dsk/c0d0p0`)。*number-of-sectors* オペランドは、デバイス上のブロック数です。これを省略すると、`mkntfs` はファイルシステムのサイズを自動的に計算します。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

次に示すように、オプションは基本、詳細、出力、およびヘルプに分類されます。

基本オプション

`-C`、`--enable-compression`
: ボリューム上で圧縮を有効にします。

`-f`、`--fast` または `-q`, `--quick`
: 簡易 (高速) フォーマットを実行します。このオプションを指定すると、ボリュームのゼロ初期化および不良セクターの検査が省略されます。

`-L`、`--label` *string*
: ファイルシステムのボリュームラベルを *string* に設定します。

`-n`、`--no-action`
: 実際にファイルシステムを作成することなく、ファイルシステムを作成する場合に実行される内容を表示します。デバイスへの実際の書き込みを除く、すべてのフォーマット処理が実行されます。

詳細オプション

`-c`、`--cluster-size` *bytes*
: クラスタのサイズをバイト単位で指定します。有効なクラスタサイズ値は 2 の累乗で、クラスタごとに 256 バイト以上 65536 バイト以下です。省略すると、デフォルトのクラスタサイズとして 4096 バイトが使用されます。

  クラスタをセクターより小さくすることはできないため、デフォルトのクラスタサイズはセクターサイズと等しいか、それより大きい値に設定してください。また、値を 4096 より大きくすると、ボリューム上で圧縮が無効になるという副作用もあります。これは、Windows で使用される NTFS 圧縮アルゴリズムの制限のためです。

`-F`、`--force`
指定したデバイスがブロック型特殊デバイスでなかったり、マウントされていると表示されている場合でも、`mkntfs` を強制的に実行します。

`-H`、`--heads` *num*
ヘッド数を指定します。最大値は 65535 (0xffff) です。省略すると、`mkntfs` が自動的にヘッド数を決定しようとします。それに失敗すると、デフォルトの 0 が使用されます。*num* の指定は、作成したボリュームから Windows をブートするために必須です。

`-I`、`--no-indexing`
ボリューム上でのコンテンツのインデックス作成を無効にします。このオプションは、Windows 2000 以降でのみ有効です。Windows NT 4.0 以前ではコンテンツのインデックス作成は実装されていないため、これは無視されます。

`-p`、`--partition-start` *sector*
パーティションの開始セクターを指定します。最大値は 4294967295 (2^{32-1}) です。省略すると、`mkntfs` が自動的に *sector* を決定しようとします。それに失敗すると、デフォルトの 0 が使用されます。*sector* の指定は、作成したボリュームから Windows をブートするために必須です。

`-S`、`--sectors-per-track` *num*
トラックあたりのセクター数を指定します。最大値は 65535 (0xffff) です。省略すると、mkntfs が自動的に sectors-per-track 数を決定しようとし、それに失敗するとデフォルトの 0 が使用されます。sectors-per-track の指定は、作成したボリュームから Windows をブートするために必須です。

`-s`、`--sector-size` *bytes*
セクターのサイズをバイト単位で指定します。有効なセクターサイズ値は 256、512、1024、2048、および 4096 です。省略すると、`mkntfs` が自動的に sector-size を決定しようとします。それに失敗すると、デフォルトのセクターあたり 512 バイトが使用されます。

`-T`、`--zero-time`
現在のシステム時間ではなく、1970 年 1 月 1 日の 00:00:00 UTC であるように見せかけます。これは、デバッグ時に役立ちます。

`-z`、`--mft-zone-multiplier` *num*
マスターファイルテーブル (MFT) ゾーンの乗数を設定します。これにより、ボリュームで使用する MFT ゾーンのサイズが決定されます。MFT ゾーンは、MFT 用に予約されているボリュームの先頭領域で、オンディスク i ノード (MFT レコード) を格納します。小さなファイルはその全体が i ノード内に格納されます。このため、非常に小さなファイルを大量にボリュームに格納する場合は、ゾーンの乗数を大きな値に設定することをお勧めします。NTFS ドライバの処理時に、MFT ゾーンのサイズは必要に応じてただちに変更されるため、オプションの値を選択することで断片化を抑えられます。有効な値は、`1`、`2`、`3`、および `4` です。各値は次の意味を持ちます。

```
MFT zone        MFT zone size
multiplier      (% of volume size)
1       12.5% (default)
2       25.0%
3       37.5%
4       50.0%
```

出力オプション

`--debug`
: `-v` オプションによる詳細出力、および `mkntfs` のデバッグに役立つ追加の出力を含めます。

`-q`、`--quiet`
: 詳細実行。エラーは標準エラー出力に書き込まれ、標準出力へは一切出力されません。これは、スクリプト内で `mkntfs` を実行する場合に役立ちます。

`-v`、`--verbose`
: 詳細実行。

ヘルプオプション

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-l`、`--license`
: `mkntfs` のライセンス情報を表示して終了します。

`-V`、`--version`
: `mkntfs` のバージョン番号を表示して終了します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

`ntfsprogs(1M)`, `ntfsresize(1M)`, `ntfsundelete(1M)`, `attributes(5)`

http://wiki.linux-ntfs.org

著者

`mkntfs` は、Anton Altaparmakov、Richard Russon、Erik Sornes、および Szabolcs Szakacsits が作成しました。

名前

mkpwdict – maintain password-strength checking database

形式

```
/usr/sbin/mkpwdict [-s dict1,... ,dictN]
      [-d destination-path]
```

機能説明

The `mkpwdict` command adds words to the dictionary-lookup database used by `pam_authtok_check(5)` and `passwd(1)`.

Files containing words to be added to the database can be specified on the command-line using the `-s` flag. These source files should have a single word per line, much like `/usr/share/lib/dict/words`.

If `-s` is omitted, `mkpwdict` will use the value of `DICTIONLIST` specified in `/etc/default/passwd` (see `passwd(1)`).

The database is created in the directory specified by the `-d` option. If this option is omitted, `mkpwdict` uses the value of `DICTIONDBDIR` specified in `/etc/default/passwd` (see `passwd(1)`). The default location is `/var/passwd`.

オプション

The following options are supported:

`-s` Specifies a comma-separated list of files containing words to be added to the dictionary-lookup database.

`-d` Specifies the target location of the dictionary-database.

ファイル

`/etc/default/passwd`	See `passwd(1)`.
`/var/passwd`	default destination directory

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`passwd(1)`, `attributes(5)`, `pam_authtok_check(5)`

名前

modinfo – ロードされているカーネルモジュールについての情報の表示

形式

`/usr/sbin/modinfo [-c] [-w] [-i module-id]`

機能説明

`modinfo` ユーティリティは、ロードされているモジュールに関する情報を表示します。表示形式は次のとおりです。

Id Loadaddr Size Info Rev Module Name

Id はモジュール ID、*Loadaddr* はテキスト開始位置のアドレス (16 進表記)、*Size* はテキストとデータと bss のバイト単位の合計サイズ (16 進表記)、*Info* はモジュール固有の情報、*Rev* はロード可能モジュールシステムのリビジョン番号、そして *Module* Name はモジュールのファイル名と内容の説明を表します。

モジュール固有の情報として表示される内容は、モジュールの種類により異なります。すなわち、ドライバであればブロックメジャー番号とキャラクタメジャー番号、システムコールであればシステムコール番号、その他の種類は指定されていません。

オプション

次のオプションがサポートされています。

`-c`	ロードされているモジュールのインスタンス数と、モジュールの現在の状態を表示します。
`-i` *module-id*	指定したモジュールに関する情報のみを表示します。
`-w`	モジュールに関する情報を、`80` 文字の位置で切り落としません。

使用例

例 1 モジュールのステータスの表示

次の例では、モジュール 2 に関する情報を表示しています。

```
example% modinfo -i 2
Id   Loadaddr   Size Info Rev Module Name
 2    ff08e000   1734   -   1  swapgeneric (root and swap configuration)
```

例 2 カーネルモジュールのステータスの表示

次の例では、いくつかのカーネルモジュールのステータスを表示しています。

```
example% modinfo
Id Loadaddr   Size Info Rev Module Name
 2  ff08e000   1734   -   1  swapgeneric
 4  ff07a000   3bc0   -   1  specfs (filesystem for specfs)
 6  ff07dbc0   2918   -   1  TS (time sharing sched class)
 7  ff0804d8    49c   -   1  TS_DPTBL (Time sharing dispatch table)
 8  ff04a000  24a30   2   1  ufs (filesystem for ufs)
 9  ff080978   c640 226   1  rpcmod (RPC syscall)
 9  ff080978   c640   -   1  rpcmod (rpc interface str mod)
10  ff08cfb8  2031c   -   1  ip (IP Streams module)
10  ff08cfb8  2031c   2   1  ip (IP Streams device)
```

例 3 -c オプションの使用

-c オプションを指定して modinfo コマンドを実行すると、読み込まれているモジュールのインスタンスの数と、そのモジュールの現在の状態が表示されます。

```
example% modinfo -c
Id    Loadcnt Module Name                          State
  1         0 krtld                            UNLOADED/UNINSTALLED
  2         0 genunix                          UNLOADED/UNINSTALLED
  3         0 platmod                          UNLOADED/UNINSTALLED
  4         0 SUNW,UltraSPARC-IIi              UNLOADED/UNINSTALLED
  5         0 cl_bootstrap                     UNLOADED/UNINSTALLED
  6         1 specfs                           LOADED/INSTALLED
  7         1 swapgeneric                      UNLOADED/UNINSTALLED
  8         1 TS                               LOADED/INSTALLED
  9         1 TS_DPTBL                         LOADED/INSTALLED
 10         1 ufs                              LOADED/INSTALLED
 11         1 fssnap_if                        LOADED/INSTALLED
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

modload(1M), modunload(1M), attributes(5)

名前

modload – カーネルモジュールのロード

形式

`modload [-p] [-e` *exec_file*`] `*filename*

機能説明

`modload` コマンドは、*filename* で示すロード可能モジュールを、稼動中のシステムにロードします。

filename は、`ld -r` で生成されたオブジェクトファイルを示します。*filename* が絶対パス名の場合、 その絶対パスが指定するファイルがロードされます。*filename* の先頭文字がスラッシュ (/) ではない場合、`-p` オプションが指定されていないと、現在のディレクトリに相対するパスを使って *filename* をロードします。

カーネルのモジュールパス `modpath` 変数は、`/etc/system` ファイルを使って設定できます。この変数のデフォルト値は、オペレーティングシステムをロードした時のパスです。通常は、`/kernel /usr/kernel` となります。

たとえば、次のように入力した場合、カーネルは `./drv/foo` を探します。

```
example# modload drv/foo
```

また、 次のように入力した場合には、カーネルはまず `/kernel/drv/foo` を探し、存在しなければ `/usr/kernel/drv/foo` を探します。

```
example# modload -p drv/foo
```

オプション

次のオプションがサポートされています。

`-e` *exec_file*
: モジュールのロードが正常に終了した後で実行すべきシェルスクリプトまたは実行可能イメージの名前を指定します。第 1 引数はモジュール ID (10 進数) です。他の引数は、モジュールの種類により異なります。各モジュールの詳細は次のとおりです。 ドライバにはブロックメジャー番号とキャラクタメジャー番号、システムコールにはシステムコール番号、その他のモジュールタイプにはそれぞれにあったカーネルテーブルへのインデックスが渡されます。`modinfo(1M)` を参照してください。

`-p`
: モジュールを検索するパスとして、カーネルの内部 modpath 変数を使用します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

`ld(1)`, `add_drv(1M)`, `kernel(1M)`, `modinfo(1M)`, `modunload(1M)`, `system(4)`, `attributes(5)`, `modldrv(9S)`, `modlinkage(9S)`, `modlstrmod(9S)`, `module_info(9S)`

『Writing Device Drivers』

注意事項

デバイスドライバを追加するには、`modload` ではなく `add_drv(1M)` を使います。デバイスドライバの追加方法については、『『Writing Device Drivers』』を参照してください。

名前

modunload – モジュールのアンロード

形式

`modunload -i` *module_id* `[-e` *exec_file*`]`

機能説明

`modunload` は、稼動中のシステムからロード可能モジュールをアンロードします。*module_id* はアンロードするモジュールの ID で、これは `modinfo(1M)` の出力情報中に得られる値と同じです。`ID` として `0` を指定した場合、自動ローディングされたモジュールのうちアンロード可能なものがすべてアンロードされます。`modload(1M)` を使ってロードしたモジュールは対象とはなりません。

オプション

次のオプションがサポートされています。

`-e` *exec_file*
: モジュールをアンロードする前に実行すべきシェルスクリプトまたは実行可能イメージの名前を指定します。第 1 引数はモジュール ID (10 進数) です。他の 2 つの引数は、モジュールの種類により異なります。ロード可能ドライバの場合は、ブロックメジャー番号が第 2 引数です。ロード可能システムコールの場合は、システムコール番号が第 2 引数です。ロード可能 exec クラスの場合は、`execsw` テーブルへのインデックスが第 2 引数です。ロード可能ファイルシステムの場合は、`vfssw` テーブルへのインデックスが第 2 引数です。ロード可能ストリームモジュールの場合は、`fmodsw` テーブルへのインデックスが第 2 引数です。ロード可能スケジューリングクラスの場合は、クラス配列へのインデックスが第 2 引数です。該当しない引数の値としては、–1 が渡されます。

`-i` *module_id*
: アンロードすべきモジュールを指定します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

`modinfo(1M)`, `modload(1M)`, `update_drv(1M)`, `attributes(5)`

注意事項

`modunload` コマンドは、関連するドライバ構成ファイルをシステムに再度読み込ませるためにドライバモジュールでよく使用されます。これは現在の Solaris リリースで機能しますが、構成ファイルを再度読み込むために勧められている方法ではなく、将来のリリースで機能する保証はありません。ドライバ構成ファイルの再読み込みには、`update_drv(1M)` コマンドを使用することをお勧めします。

名前

mofcomp – compile MOF files into CIM classes

形式

```
/usr/sadm/bin/mofcomp [-c cimom_hostname ] [-h]
     [-j filename] [-n namespace] [-o dirname]
     [-p password ] [-CIQ] [-u username] [-v ] [-version]
     [-x] file
```

機能説明

The `mofcomp` utility is executed during installation to compile managed object format (MOF) files that describe the Common Information Model (CIM) and Solaris Schemas into the CIM Object Manager Repository, a central storage area for management data. The CIM Schema is a collection of class definitions used to represent managed objects that occur in every management environment. The Solaris Schema is a collection of class definitions that extend the CIM Schema and represent managed objects in a typical Solaris operating environment.

The `mofcomp` utility must be run as root or as a user with write access to the namespace in which you are compiling.

MOF is a language for defining CIM classes and instances. MOF files are ASCII text files that use the MOF language to describe CIM objects. A CIM object is a computer representation or model of a managed resource, such as a printer, disk drive, or CPU.

Many sites store information about managed resources in MOF files. Because MOF can be converted to Java, Java applications that can run on any system with a Java Virtual Machine can interpret and exchange this information. You can also use the `mofcomp` utility to compile MOF files at any time after installation.

オプション

The following options are supported:

`-c` *cimom_hostname*	Specify a remote system running the CIM Object Manager.
`-C`	Run the compiler set with the class option, which updates a class if it exists, and returns an error if the class does not exist. If you do not specify this option, the compiler adds a CIM class to the connected namespace, and returns an error if the class already exists.
`-h`	List the arguments to the `mofcomp` utility.
`-I`	Run the compiler set with the instance option, which updates an instance if it exists, and returns an error if the instance does not exist. If you do not specify this option, the compiler adds a CIM instance to the connected namespace, and returns an error if the instance already exists.
`-j` *filename*	Generate Java Beans and Java Interfaces to manage the CIM instances related to the CIM classes in the MOF being compiled. The contents of *filename* are: `PACKAGE=`*Java package name* `IMPORTS=`*import1*:...:*importN*

<EXCEPTIONS=*exception1*:...:*exceptionN*

PACKAGE is a valid Java package name to include in all generated Java source. IMPORTS is an optional colon separated list of valid Java classes to be imported in all generated Java source. EXCEPTIONS is an optional colon separated list of valid Java exceptions to be thrown by the methods in all generated Java source.

-n *namespace*

Requests that the compiler load the MOF file into the namespace specified as *namespace*. The default namespace (root\cimv2) is used unless this switch is used or a #pragma *namespace* ("namespace") statement appears in the MOF file. If both the -n *namespace* switch and the #pragma *namespace* construct are used, all namespaces are created, but the objects are created only in the #pragma namespaces.

-o *dirname*

Run compiler in standalone mode, without the CIM Object Manager. Specify dirname as the directory in which the compiler output is to be stored. In this mode, the CIM Object Manager need not be running.

-p *password*

Specify a password for connecting to the CIM Object Manager. Use this option for compilations that require privileged access to the CIM Object Manager. If you specify both -p and -u, you must type the password on the command line, which can pose a security risk. A more secure way to specify a password is to specify -u but not -p, so that the compiler will prompt for the password.

-Q

Run the compiler set with the qualifier types option, which updates a qualifier type if it exists, and returns an error if the qualifier type does not exist. If you do not specify this option, the compiler adds a CIM qualifier type to the connected namespace, and returns an error if the qualifier type already exists.

-u *username*

Specify user name for connecting to the CIM Object Manager. Use this option for compilations that require privileged access to the CIM Object Manager. If you specify both -p and -u, you must type the password on the command line, which can pose a security risk. A more secure way to specify a password is to specify -u but not -p, so that the compiler will prompt for the password.

-v

Run the compiler in verbose mode, which displays compiler messages.

-version

Display the version of the MOF compiler.

-x

Generate XML documents for the CIM classes defined in the input MOF file.

オペランド

The following operands are supported:

file The pathname of the file to be compiled.

終了ステータス

The `mofcomp` utility exits with `0` upon success and a positive integer upon failure.

ファイル

MOF files are installed in `/usr/sadm/mof`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

`init.wbem(1M)`, `mofreg(1M)`, `wbemadmin(1M)`, `wbemlogviewer(1M)`, `attributes(5)`, `wbem(5)`

名前

mofreg – register MOF classes with WBEM services

形式

```
/usr/sadm/bin/mofreg -r tag file
/usr/sadm/bin/mofreg -s
/usr/sadm/bin/mofreg -u tag [file]
```

機能説明

The `mofreg` command is used by package and patch install scripts, or by any applications that wish to register managed object format (MOF) classes with Sun The Web-Based Enterprise Management (WBEM) services.

The WBEM services daemon (Common Information Model or CIM object manager) processes at start up the files that are specified by `mofreg` commands. Files are processed in the order that the individual `mofreg` commands are executed.

As an alternative to using the `mofreg` command, MOFs can be registered or unregistered by manipulating directories in `/var/sadm/wbem/logr`. Instead of running the `mofreg -r tag file` version fo the command you can create a directory named `tag` under `/var/sadm/wbem/logr/preReg` and copy `file` to the `tag` directory.

Similarly, instead of running the `mofreg -u tag [file]` command, you can create a directory named `tag` under `/var/sadm/wbem/logr/preUnreg` and copy the optional `file` to the `tag` directory.

The entries are processed in increasing order of last modification time of the `tag` directories. If you issue `mofreg` commands in rapid succession, the timestamps might be the same. If you have a situation where the timestamp order is critical, you can place appropriate sleeps between the successive registration or unregistration operations. As with the `mofreg` command, processing is done at next restart or by using the `-s` option.

This alternative mechanism is typically used in package install scripts which do not have access to `/usr`, and therefore do not have access to the `mofreg` command. This case arises when packages are installed for diskless clients.

オプション

The following options are supported:

`-r` *tag file*

The *file* argument is the actual MOF registration file. Its form is identical to the MOF syntax as defined by the Distributed Management Task Force (DMTF). The only difference is the addition of the following 3 new pseudo-pragmas, which are variations of the namespace pragma. The name of file cannot end in `.unreg`.

```
#pragma namespace("__create")
#pragma namespace("__delete")
#pragma namespace("__modify")
```

These three pragmas are used specify if the elements following the pragmas should be created, deleted, or modified by the CIM object manager. The `__delete` pragma can currently only be applied for a `mofreg -u` command.

The *tag* argument is a unique string that specifies the identity of the registry action. This tag can be set to the package name or the patch number if the `mofreg` script is being invoked through packages/patches, though any tag can be specified.

Errors and warnings that are encountered when the CIM object manager handles the `mofreg` script are logged. Processing of the `mofreg` script stops at the first error. Specific warnings include:

```
Element already defined - the element already exists and
cannot be created.

Element not found - the element does not exist and
cannot be modified.
```

The error conditions are:

```
Key modification - A class cannot be modified if its keys
are being changed.

Other mod compilation errors.
```

`-s` Forces the CIM object manager to immediately process outstanding registry requests, instead of at the next restart. This currently requires Java.

`-u` *tag* [*file*] Undoes the operations performed during `mof` registry.

The *tag* argument must correspond to the value set during the original `mofreg` invocation. If no `mofreg` was done with the original *tag*, the command does not succeed.

If required, an *unreg* file can be specified. If no *unreg* file is specified, the CIM object manager automatically undoes the actions of the registry. Any class created by the registry process is removed and any classes modified by the registry revert to the old state.

The `mofreg` command does not take care of cases where packages and patches make conflicting changes to classes. This should be taken care of by the standard patch and package conflict resolution.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` An error occurred. The reason for error is displayed.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

init.wbem(1M), mofcomp(1M), wbemadmin(1M), wbemlogviewer(1M), attributes(5), wbem(5)

名前

monitor – SPARC system PROM monitor

形式

```
STOP–A

BREAK

initial system power-on

exit from a client program, e.g., the Operating System
```

機能説明

The CPU board of a workstation contains one or more EPROMs or EEPROMs. The program which executes from the PROMs is referred to as “the monitor”. Among other things, the monitor performs system initialization at power-on and provides a user interface.

Monitor Prompt

The monitor of earlier workstations was known as the `SunMON` monitor and displayed the > for its prompt. See the `SunMON MONITOR USAGE` section for further details.

Existing workstations use a monitor which is known as the OpenBoot monitor. The OpenBoot monitor typically displays `ok` as its prompt, but it may also display the > prompt under certain circumstances.

If the `'auto-boot?'` NVRAM parameter is set to `'false'` when the workstation is powered on, the system does not attempt to boot and the monitor issues its prompt. If `'auto-boot'` is set to `'true'`, the system initiates the boot sequence. The boot sequence can be aborted by simultaneously pressing two keys on the system's keyboard: L1 and A (on older keyboards), or Stop and A (on newer keyboards). Either a lower case `a` or an upper case `A` works for the keyboard abort sequence. If a console has been attached by way of one of the system's serial ports then the abort sequence can be accomplished by sending a BREAK. See `tip(1)`.

When the NVRAM `'security-mode'` parameter has been turned on, or when the value of the `'sunmon-compat?'` parameter is true, then the OpenBoot monitor displays the message: `Type b (boot), c (continue), or n (new command mode)`

and the > prompt appears.

Openboot Prom Usage

Some of the more useful commands that can be issued from OpenBoot's `ok` prompt are described here. Refer to the 『OpenBoot 2.x コマンド・リファレンスマニュアル』 book for a complete list of commands.

Help

Help for various functional areas of the OpenBoot monitor can be obtained by typing `help`. The help listing provides a number of other key words which can then be used in the help command to provide further details.

NVRAM Parameters

Each workstation contains one or more NVRAM devices which contains unique system ID information, as well as a set of user-configurable parameters. The NVRAM parameters allow the user a certain level of flexibility in configuring the system to act in a given manner under a specific set of circumstances.

See `eeprom(1M)` for a description of the parameters and information regarding setting the parameters from the OS level.

The following commands can be used at the OpenBoot monitor to access the NVRAM parameters.

`printenv`	Used to list the NVRAM parameters, along with their default values and current values.
`setenv` *pn pv*	Used to set or modify a parameter. The *pn* represents the parameter name, and *pv* represents the parameter value.
`set-default` *pn*	Used to set an individual parameter back to its default value.
`set-defaults`	Used to reset all parameters to their default values. (Note that `'set-defaults'` only affects parameters that have assigned default values.)

Security Parameters

Newer OpenBoot monitors contain user interfaces that support the storage and listing of keys for later use by client programs.

`list-security-keys`	Lists the names of keys currently stored on a machine.
`set-security-key` *keyname* [*keydata*]	Stores key data *keydata* in a key named *keyname*. Actual key data can be up to 32 bytes in length. The maximum length of *keyname* is 64 bytes, which allows for the hex-formatted ASCII used to present the key data. If *keydata* is not present, *keyname* and its corresponding data is deleted.

Hardware Checks and Diagnostics

The following commands are available for testing or checking the system's hardware. If the `'diag-switch?'` NVRAM parameter is set to true when the system is powered on, then a Power-On Self Test (POST) diagnostic is run, if present, sending its results messages to the system's serial port A. Not all of the commands shown are available on all workstations.

`test-all`	Run the diagnostic tests on each device which has provided a self-test.
`test /memory`	Run the main memory tests. If the NVRAM parameter `'diag-switch?'` is set to true, then all of main memory is tested. If the parameter is false then only the amount of memory specified in the `'selftest-#megs'` NVRAM parameter is tested.
`test net`	Test the network connection for the on-board network controller.
`watch-net`	Monitor the network attached to the on-board net controller.
`watch-net-all`	Monitor the network attached to the on-board net controller, as well as the network controllers installed in SBus slots.
`watch-clock`	Test the system's clock function.

System Information

The following commands are available for displaying information about the system. Not all commands are available on all workstations.

`banner`	Display the power-on banner.
`.enet-addr`	Display the system's Ethernet address.
`.idprom`	Display the formatted contents of the IDPROM.
`module-info`	Display information about the system's processor(s).
`probe-scsi`	Identify the devices attached to the on-board SCSI controller.
`probe-scsi-all`	Identify the devices attached to the on-board SCSI controller as well as those devices which are attached to SBus SCSI controllers.
`show-disks`	Display a list of the device paths for installed SCSI disk controllers.
`show-displays`	Display a list of the device paths for installed display devices.
`show-nets`	Display a list of the device paths for installed Ethernet controllers.
`show-sbus`	Display list of installed SBus devices.
`show-tapes`	Display a list of the device paths for installed SCSI tape controllers.
`show-ttys`	Display a list of the device paths for tty devices.
`.traps`	Display a list of the SPARC trap types.
`.version`	Display the version and date of the OpenBoot PROM.

Emergency Commands

These commands must be typed from the keyboard, they do not work from a console which is attached by way of the serial ports. With the exception of the Stop-A command, these commands are issued by pressing and holding down the indicated keys on the keyboard immediately after the system has been powered on. The keys must be held down until the monitor has checked their status. The Stop-A command can be issued at any time after the console display begins, and the keys do not need to be held down once they've been pressed. The Stop-D, Stop-F and Stop-N commands are not allowed when one of the security modes has been set. Not all commands are available on all workstations.

Stop (L1)	Bypass the Power-On Self Test (POST). This is only effective if the system has been placed into the diagnostic mode.
Stop-A (L1-A)	Abort the current operation and return to the monitor's default prompt.
Stop-D (L1-D)	Set the system's `'diag-switch?'` NVRAM parameter to `'true'`, which places the system in diagnostic mode. POST diagnostics, if present, are run, and the messages are displayed by way of the system's serial port A.
Stop-F (L1-F)	Enter the OpenBoot monitor before the monitor has probed the system for devices. Issue the `'fexit'` command to continue with system initialization.

Stop-N (L1-N)	Causes the NVRAM parameters to be reset to their default values. Note that not all parameters have default values.

Line Editor Commands

The following commands can be used while the monitor is displaying the `ok` prompt. Not all of these editing commands are available on all workstations.

CTRL-A	Place the cursor at the start of line.
CTRL-B	Move the cursor backward one character.
ESC-B	Move the cursor backward one word.
CTRL-D	Erase the character that the cursor is currently highlighting.
ESC-D	Erase the portion of word from the cursor's present position to the end of the word.
CTRL-E	Place the cursor at the end of line.
CTRL-F	Move the cursor forward one character.
ESC-F	Move the cursor forward one word.
CTRL-H	Erase the character preceding the cursor (also use Delete or Back Space)
ESC-H	Erase the portion of the word which precedes the cursor (use also CTRL-W)
CTRL-K	Erase from the cursor's present position to the end of the line.
CTRL-L	Show the command history list.
CTRL-N	Recall the next command from the command history list
CTRL-P	Recall a previous command from the command history list.
CTRL-Q	Quote the next character (used to type a control character).
CTRL-R	Retype the current line.
CTRL-U	Erase from the cursor's present position to the beginning of the line.
CTRL-Y	Insert the contents of the memory buffer into the line, in front (to the left) of the cursor.

nvramrc

The `nvramrc` is an area of the system's NVRAM where users may store Forth programs. The programs which are stored in the `nvramrc` are executed each time the system is reset, provided that the `'use-nvramrc?'` NVRAM parameter has been set to `'true'`. Refer to the 『OpenBoot 2.x コマンド・リファレンスマニュアル』 book for information on how to edit and use the `nvramrc`.

Restricted Monitor

The command `'old-mode'` is used to move OpenBoot into a restricted monitor mode, causing the > prompt to be displayed. Only three commands are allowed while in the restricted monitor; the `'go'` command (to resume a program which was interrupted with the

Stop-A command), the 'n' command (to return to the normal OpenBoot monitor), and boot commands. The restricted monitor's boot commands approximate the older SunMON monitor's boot command syntax. If a 'security-mode' has been turned on then the restricted monitor becomes the default monitor environment. The restricted monitor may also become the default environment if the 'sunmon-compat?' NVRAM parameter is set to true. Not all workstations have the 'sunmon-compat?' parameter.

SunMON Prom Usage

The following commands are available systems with older SunMON-based PROM:

+|–
: Increment or decrement the current address and display the contents of the new location.

^C *source destination n*
: (caret-C) Copy, byte-by-byte, a block of length *n* from the source address to the *destination* address.

^I *program*
: (caret-I) Display the compilation date and location of *program*.

^T *virtual_address*
: (caret-T) Display the physical address to which *virtual_address* is mapped.

b [!] [*device* [(*c*,*u*,*p*)]] [*pathname*] [*arguments_list*]

b[?]
: Reset appropriate parts of the system and bootstrap a program. A '!' (preceding the *device* argument) prevents the system reset from occurring. Programs can be loaded from various devices (such as a disk, tape, or Ethernet). 'b' with no arguments causes a default boot, either from a disk, or from an Ethernet controller. 'b?' displays all boot devices and their *devices*.

device	one of	
	le	Lance Ethernet
	ie	Intel Ethernet
	sd	SCSI disk, CDROM
	st	SCSI 1/4– or 1/2–inch tape
	fd	Diskette
	id	IPI disk
	mt	Tape Master 9-track 1/2–inch tape
	xd	Xylogics 7053 disk
	xt	Xylogics 1/2–inch tape
	xy	Xylogics 440/450 disk

c	A controller number (`0` if only one controller),
u	A unit number (`0` if only one driver), and
p	A partition.
pathname	A pathname for a program such as `/stand/diag`.
arguments_list	A list of up to seven arguments to pass to the program being booted.

`c` [*virtual_address*]
Resume execution of a program. When given, *virtual_address* is the address at which execution resumes. The default is the current PC. Registers are restored to the values shown by the `d`, and `r` commands.

`d` [*window_number*]
Display (dump) the state of the processor. The processor state is observable only after:

- An unexpected trap was encountered.
- A user program dropped into the monitor (by calling *abortent*).
- The user manually entered the monitor by typing L1–A or BREAK.

The display consists of the following:

- The special registers: PSR, PC, nPC, TBR, WIM, and Y
- Eight global registers
- 24 window registers (8 *in*, 8 *local*, and 8 *out*), corresponding to one of the 7 available windows. If a Floating-Point Unit is on board, its status register along with 32 floating-point registers are also shown.

window_number	Display the indicated *window_number*, which can be any value between `0` and `6`, inclusive. If no window is specified and the PSR's current window pointer contains a valid window number, registers from the window that was active just prior to entry into the monitor are displayed. Otherwise, registers from window 0 are displayed.

`e` [*virtual_address*] [*action*] . . .
Open the 16-bit word at *virtual_address* (default zero). The address is interpreted in the address space defined by the `s` command. See the `a` command for a description of *action*.

`f` *virtual_address1 virtual_address2 pattern* [*size*]
Fill the bytes, words, or long words from *virtual_address1* (lower) to *virtual_address2* (higher) with the constant, *pattern*. The `size` argument can take one of the following values:

`b`	byte format (the default)
`w`	word format
`l`	long word format

For example, the following command fills the address block from `0x1000` to `0x2000` with the word pattern, `0xABCD`:

```
f 1000 2000 ABCD W
```

`g` [*vector*] [*argument*]
`g` [*virtual_address*] [*argument*]
Goto (jump to) a predetermined or default routine (first form), or to a user-specified routine (second form). The value of *argument* is passed to the routine. If the *vector* or *virtual_address* argument is omitted, the value in the PC is used as the address to jump to.

To set up a predetermined routine to jump to, a user program must, prior to executing the monitor's `g` command, set the variable `*romp->v_vector_cmd` to be equal to the virtual address of the desired routine. Predetermined routines need not necessarily return control to the monitor.

The default routine, defined by the monitor, prints the user-supplied *vector* according to the format supplied in *argument*. This format can be one of:

`%x` hexadecimal

`%d` decimal

`g0`
Force a panic and produce a crash dump when the monitor is running as a result of the system being interrupted,

`g4`
(Sun-4 systems only) Force a kernel stack trace when the monitor is running as a result of the system being interrupted,

`h`
Display the help menu for monitor commands and their descriptions. To return to the monitor's basic command level, press ESCAPE or `q` before pressing RETURN.

`i` [*cache_data_offset*] [*action*] . . .
Modify cache data RAM command. Display and/or modify one or more of the cache data addresses. See the `a` command for a description of *action*.

`j` [*cache_tag_offset*] [*action*] . . .
Modify cache tag RAM command. Display and/or modify the contents of one or more of the cache tag addresses. See the `a` command for a description of *action*.

`k` [*reset_level*]
Reset the system, where *reset_level* is:

`0` Reset VMEbus, interrupt registers, video monitor (Sun-4 systems). This is the default.

`1` Software reset.

2 Power-on reset. Resets and clears the memory. Runs the EPROM-based diagnostic self test, which can take several minutes, depending upon how much memory is being tested.

`kb`
Display the system banner.

`l` [*virtual_address*] [*action*] . . .
Open the long word (32 bit) at memory address *virtual_address* (default zero). The address is interpreted in the address space defined by the `s` command (below). See the `a` command for a description of *action*.

`m` [*virtual_address*] [*action*] . . .
Open the segment map entry that maps *virtual_address* (default zero). The address is interpreted in the address space defined by the `s` command. See the `a` command for a description of *action*.

`ne`

`ni`
Disable, enable, or invalidate the cache, respectively.

`o` [*virtual_address*] [action*]* . . .
Open the byte location specified by *virtual_address* (default zero). The address is interpreted in the address space defined by the `s` command. See the `a` command for a description of *action*.

`p` [*virtual_address*] [*action*]. . .
Open the page map entry that maps *virtual_address* (default zero) in the address space defined by the `s` command. See the `a` command for a description of *action*.

`q` [*eeprom_offset*] [*action*]. . .
Open the EEPROM *eeprom_offset* (default zero) in the EEPROM address space. All addresses are referenced from the beginning or base of the EEPROM in physical address space, and a limit check is performed to insure that no address beyond the EEPROM physical space is accessed. This command is used to display or modify configuration parameters, such as: the amount of memory to test during self test, whether to display a standard or custom banner, if a serial port (A or B) is to be the system console, etc. See the `a` command for a description of *action*.

`r` [*register_number*]
`r` [*register_type*]
`r` [*w window_number*]
Display and/or modify one or more of the IU or FPU registers. A hexadecimal *register_number* can be one of:

`0x00–0x0f`	window(0,i0)–window(0,i7), window(0,i0)—window(0,i7)
`0x16–0x1f`	window(1,i0)–window(1,i7), window(1,i0)—window(1,i7)

`0x20–0x2f`	window(2,i0)–window(2,i7), window(2,i0)—window(2,i7)
`0x30–0x3f`	window(3,i0)–window(3,i7), window(3,i0)—window(3,i7)
`0x40–0x4f`	window(4,i0)–window(4,i7), window(4,i0)—window(4,i7)
`0x50–0x5f`	window(5,i0)–window(5,i7), window(5,i0)—window(5,i7)
`0x60–0x6f`	window(6,i0)–window(6,i7), window(6,i0)—window(6,i7)
`0x70–0x77`	`g0, g1, g2, g3, g4, g5, g6, g7`
`0x78–0x7d`	PSR, PC, nPC, WIM, TBR, Y.
`0x7e–0x9e`	FSR, f0–f31

Register numbers can only be displayed after an unexpected trap, a user program has entered the monitor using the *abortent* function, or the user has entered the monitor by manually typing L1–A or BREAK.

If a *register_type* is given, the first register of the indicated type is displayed. *register_type* can be one of:

`f`	floating-point
`g`	global
`s`	special

If `w` and a *window_number* (`0`—`6`) are given, the first *in*-register within the indicated window is displayed. If *window_number* is omitted, the window that was active just prior to entering the monitor is used. If the PSR's current window pointer is invalid, window 0 is used.

`s` [*asi*])

Set or display the Address Space Identifier. With no argument, `s` displays the current Address Space Identifier. The *asi* value can be one of:

`0x2`	control space
`0x3`	segment table
`0x4`	Page table
`0x8`	user instruction
`0x9`	supervisor instruction
`0xa`	user data
`0xb`	supervisor data
`0xc`	flush segment

`0xd` flush page

`0xe` flush context

`0xf` cache data

`u` [`echo`]

`u` [*port*] [*options*] [*baud_rate*]

`u` [`u`] [*virtual_address*]

With no arguments, display the current I/O device characteristics including: current input device, current output device, baud rates for serial ports A and B, an input-to-output echo indicator, and virtual addresses of mapped UART devices. With arguments, set or configure the current I/O device. With the `u` argument (`uu`. . .), set the I/O device to be the *virtual_address* of a UART device currently mapped.

`echo` Can be either `e` to enable input to be echoed to the output device, or `ne`, to indicate that input is not echoed.

port Assign the indicated *port* to be the current I/O device. *port* can be one of:

- `a` serial port A
- `b` serial port B
- `k` the workstation keyboard
- `s` the workstation screen

baud_rate Any legal baud rate.

options can be any combination of:

- `i` input
- `o` output
- `u` UART
- `e` echo input to output
- `ne` do not echo input
- `r` reset indicated serial port (`a` and `b` ports only)

If either `a` or `b` is supplied, and no *options* are given, the serial port is assigned for both input and output. If `k` is supplied with no options, it is assigned for input only. If `s` is supplied with no options, it is assigned for output only.

`v` *virtual_address1 virtual_address2* [`size`]

Display the contents of *virtual_address1* (lower) *virtual_address2* (higher) in the format specified by `size`:

`b` byte format (the default)

`w` word format

`l` long word format

Enter return to pause for viewing; enter another return character to resume the display. To terminate the display at any time, press the space bar.

For example, the following command displays the contents of virtual address space from address `0x1000` to `0x2000` in word format:

```
v 1000 2000 W
```

`w` [*virtual_address*] [*argument*]
Set the execution vector to a predetermined or default routine. Pass *virtual_address* and *argument* to that routine.

To set up a predetermined routine to jump to, a user program must, prior to executing the monitor's `w` command, set the variable `*romp->v_vector_cmd` to be equal to the virtual address of the desired routine. Predetermined routines need not necessarily return control to the monitor.

The default routine, defined by the monitor, prints the user-supplied *vector* according to the format supplied in *argument*. This format can be one of:

`%x` hexadecimal

`%d` decimal

`x`
Display a menu of extended tests. These diagnostics permit additional testing of such things as the I/O port connectors, video memory, workstation memory and keyboard, and boot device paths.

`y c` *context_number*
Display context number.

`y p|s` *context_number virtual_address*
Flush the indicated context, context page, or context segment.

`c` flush context *context_number*

`p` flush the page beginning at *virtual_address* within context *context_number*

`s` flush the segment beginning at *virtual_address* within context *context_number*

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	SPARC

関連項目 tip(1), boot(1M), eeprom(1M), attributes(5)

『OpenBoot 2.x コマンド・リファレンスマニュアル』

名前

mount, umount – ファイルシステムとリモートリソースのマウントおよびマウント解除

形式

```
mount [-p | -v]

mount [-F FSType] [generic_options] [-o specific_options]
      [-O] special | mount_point

mount [-F FSType] [generic_options] [-o specific_options]
      [-O] special mount_point

mount -a [-F FSType] [-V] [current_options]
      [-o specific_options] [mount_point]...

umount [-f] [-V] [-o specific_options] special | mount_point

umount -a [-f] [-V] [-o specific_options] [mount_point]...
```

機能説明

mount は、ファイルシステムをファイルシステム階層内の *mount_point* (ディレクトリのパス名) に継ぎ足します。マウント操作を実行する前に *mount_point* の下に存在したファイルとディレクトリは、ファイルシステムのマウントを解除するまで見えなくなります。

umount は、現在マウントされているファイルシステムをマウント解除します。マウントを解除するファイルシステムは *mount_point* または *special* (ファイルシステムが存在しているデバイス) のどちらかで指定できます。

現在マウントされているファイルシステムのテーブルは、マウント済みファイルシステム情報ファイルにあります。このファイルは通常、/etc/mnttab 上にマウントされているファイルシステムによって提供されます。マウント済のファイルシステムについての情報は、mnttab(4) に格納されています。ファイルシステムをマウントすると、マウントテーブルにエントリが追加されます。(umount で) ファイルシステムをマウント解除すると、テーブルからエントリが削除されます。

special および *mount_point* 両方の引数と -F オプションを指定した場合、mount は *special* 以外のすべての引数の妥当性を検査し、適切な *FSType* に固有な mount モジュールを呼び出します。引数なしで呼び出した場合、mount は、マウントテーブル /etc/mnttab に記録されているすべてのマウント済みファイルシステムをリストします。引数を一部だけ、たとえば、*special* か *mount_point* のどちらか 1 つだけを指定した場合や、*special* と *mount_point* の両方を指定したが、*FSType* は指定しない場合、mount は /etc/vfstab を調べて、指定されていない引数を補完するエントリを探します。そのようなエントリが見つからず、また、special 引数が / で始まる場合、/etc/default/fs に指定されているデフォルトのローカルのファイルシステムタイプが使用されます。それ以外の場合、デフォルトのリモートファイルシステムタイプが使用されます。デフォルトのリモートファイルシステムタイプは、/etc/dfs/fstypes ファイルの最初のエントリによって決定されます。指定されていない引数を補完した後、mount は *FSType* に固有な mount モジュールを呼び出します。

サポートしているファイルシステムタイプの場合は、special 引数にファイルのフルパスを指定することで、ファイルをファイルシステムとして直接マウントできます。このような場合は、nosuid オプションが必要になります。この種のループバックファイルマウントをサポートするファイルシステムが存在しない場合でも、lofiadm(1M) を使用するとファイルシステムイメージをマウントできます。この場合、special オプションは不要です。

mount や umount を使用してファイルシステムのマウントやマウント解除を行えるのは、十分な特権 (PRIV_SYS_MOUNT 以上) を持つユーザーだけです。ただし、マウントされているファイルシステムとリソースを一覧表示するだけであれば、だれでも mount コマンドを使用することができます。

オプション

-F *FSType*
: 操作の対象となる *FSType* を指定します。*FSType* は、明示的に指定するか、あるいは、/etc/vfstab から決定できるか、/etc/default/fs または /etc/dfs/fstypes を調べることで決定できるものでなければなりません。

-a [*mount_points*...]
: 可能であれば、複数の mount または umount 操作を同時に実行します。

  マウントポイントを指定しないと、mount は /etc/vfstab において mount at boot フィールドが yes に設定されているすべてのファイルシステムをマウントします。マウントポイントを指定すると、/etc/vfstab の mount at boot フィールドは無視されます。

  マウントポイントを指定すると、umount は指定されたマウントポイントだけをマウント解除します。マウントポイントを指定しないと、umount は /etc/mnttab にあるすべてのファイルシステムのマウントを解除します。ただし、次に挙げる、システムに必須のファイルシステムは除きます。/、/usr、/var、/var/adm、/var/run、/proc、/dev/fd、および /tmp です。

-f
: ファイルシステムのマウントを強制的に解除します。

  このオプションを指定しないと、umount は、ファイルシステム上のファイルがビジー状態である場合、ファイルシステムをマウント解除しません。このオプションを指定すると、オープンしているファイルのデータが失われる可能性があります。ファイルシステムをマウント解除した後にプログラムがファイルにアクセスしようとすると、エラー (EIO) が返されます。

-p
: マウントされているファイルシステムの一覧を /etc/vfstab 形式で表示します。このオプションを指定するときは他のオプションを指定してはいけません。「使用上の留意点」を参照してください。

-v
: マウントされているファイルシステムの一覧を詳細形式で表示します。このオプションを指定するときは他のオプションを指定してはいけません。

-V
コマンド行全体をエコーしますが、コマンドは実行しません。エコーされたコマンドには、ユーザーが指定したオプションと引数、さらに、/etc/mnttab から得られた情報が追加されます。このオプションは、コマンド行を確認および検証するときに使用します。

generic_options
ほとんどの *FSType* に固有なコマンドモジュールでサポートされている共通のオプションです。次のオプションを指定できます。

-m
ファイルシステムをマウントしますが、/etc/mnttab にエントリを作成しません。

-g
ファイルシステムを広域的にマウントします。クラスタ化されているシステムでは、クラスタ内にあるすべてのノード上でファイルシステムを広域的にマウントします。クラスタ化されていないシステムでは、このオプションは何の効果もありません。

-o
サブオプションとキーワード属性の組の並を、スペースを入れずにコンマで区切った形式で、*FSType* 固有のオプションを指定します。これらのオプションはコマンドの *FSType* に固有なモジュールによって解釈されます (mount_ufs(1M) を参照)。/etc/vfstab にエントリがあるファイルシステムで -o オプションを指定すると、/etc/vfstab 内でそのファイルシステムに指定されている mount コマンドのオプションはすべて無視されます。

次のオプションがサポートされています。

devices | nodevices
デバイス固有ファイルを開くことを許可または禁止します。デフォルトは devices です。

devices と一緒に nosuid を使用すると、その動作は nosuid と同等になります。

exec | noexec
ファイルシステムでプログラムを実行することを許可または禁止します。ファイルシステム内のファイルに対して、PROT_EXEC 付きの mmap(2) を許可または禁止します。デフォルトは exec です。

loop
互換性のために無視されます。

nbmand | nonbmand
当該ファイルシステムに対して、非ブロック必須ロック意味論を許可または禁止します。デフォルトでは、非ブロック必須ロックは無効です。

ファイルシステムが nbmand オプションでマウントされている場合、アプリケーションは fcntl(2) インタフェースを使用して、非ブロック必須ロックをファイルにかけることができます。すると、システムは非ブロック必須ロック意味論を実施します。このオプションを有効にした場合、標準に適合するアプリケーションに予期せぬエラーが発生する可能性があります。

システムファイルへの強制ロックを取得しないようにするため、次のファイルシステムでは nbmand オプションを使用しないでください。

```
/
/usr
/etc
/var
/proc
/dev
/devices
/system/contract
/system/object
/etc/mnttab
/etc/dfs/sharetab
```

remount オプションを使用して、ファイルシステムの nbmand 設定を変更しないでください。nbmand オプションはグローバルオプションとは相互排他的です。-g を参照してください。

ro | rw
: 読み取り専用または読み書きを指定します。デフォルトは rw です。

setuid | nosetuid
: setuid または setgid の実行を許可または禁止します。デフォルトは setuid です。

setuid と一緒に nosuid を使用すると、その動作は nosuid と同等になります。

nosuid は、nosetuid と nodevices と同等です。suid または nosuid を setuid または nosetuid および devices または nodevices と一緒に組み合わせた場合、ほとんどの制限オプションが有効になります。

root= オプションを指定した NFS を使用して、ファイルシステムを共有している場合、このオプションを指定することを強く推奨します。このオプションを設定しないと、NFS クライアントは setuid プログラムをサーバーに追加したり、セキュリティホールを開けるデバイスを作成する可能性があります。

suid | nosuid
: setuid または setgid の実行を許可または禁止します。デフォルトは suid です。このオプションは、また、ファイルシステム内にあらわれる任意のデバイス固有エントリを開くことを許可または禁止します。

nosuid は、nosetuid と nodevices と同等です。suid または nosuid を setuid または nosetuid および devices または nodevices と一緒に組み合わせた場合、ほとんどの制限オプションが有効になります。

root= オプションを設定した NFS を使用して、ファイルシステムを共有している場合、このオプションを設定することを強く推奨します。このオプションを設定しないと、NFS クライアントは setuid プログラムをサーバーに追加したり、セキュリティホールを開けるデバイスを作成する可能性があります。

rstchown | norstchown
制限付き chown を許可または禁止します。ファイルシステムのマウント時に rstchown が指定される場合、ファイルの所有者はそのファイルの所有者 ID を変更できません。ファイルシステムのマウント時に norstchown が指定される場合、ユーザーは所有するファイルの所有権変更を許可できます。スーパーユーザーまたは適切な特権を持つユーザーだけが、所有者 ID を変更できます。

-O
オーバーレイマウント。既存のマウントポイント上にファイルシステムをマウントできます。これより既存のマウントポイントのファイルシステムにはアクセスできなくなります。このフラグを設定せずに既存のマウントポイント上にファイルシステムをマウントしようとすると、マウントは失敗して、device busy というエラーメッセージが表示されます。

-r
ファイルシステムを読み取り専用でマウントします。

使用例

例 1 DVD イメージを直接マウントおよびマウント解除する

次のコマンドでは、DVD イメージのマウントとマウント解除を行います。

```
# mount -F hsfs /images/solaris.iso /mnt/solaris-image
# umount /mnt/solaris-image
```

使用法

2G バイト (2^{31} バイト) 以上のファイルを検出した場合の mount と umount の動作については、largefile(5) を参照してください。

ファイル

/etc/mnttab
マウントされているファイルシステムのテーブル

/etc/default/fs
デフォルトのローカルのファイルシステムタイプ。デフォルト値は、/etc/default/fs 内で次のように設定されています。例: LOCAL=ufs

LOCAL:
FSType を指定しない場合に、コマンドがデフォルトで使用するパーティション

/etc/vfstab
各ファイルシステム用のデフォルトのパラメータリスト

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

lofiadm(1M), mount_hsfs(1M), mount_nfs(1M), mount_pcfs(1M), mount_smbfs(1M), mount_tmpfs(1M), mount_udfs(1M), mount_ufs(1M), mountall(1M), umountall(1M), fcntl(2), mmap(2), mnttab(4), vfstab(4), attributes(5), largefile(5), privileges(5), lofs(7FS), pcfs(7FS)

注意事項

ファイルシステムがマウントされるディレクトリがシンボリックリンクの場合、ファイルシステムは、シンボリックリンク自身ではなく、シンボリックリンクが参照するディレクトリ上にマウントされます。

使用上の留意点

mount -p の出力は、cachefs については正しくありません。

名前

mountall, umountall – 複数のファイルシステムのマウントおよびマウント解除

形式

```
mountall [-F FSType] [-l | -r] [file_system_table]

umountall [-k] [-s] [-F FSType] [-l | -r] [-n]  [-Z]

umountall [-k] [-s] [-h host] [-n] [-Z]
```

機能説明

mountall は、ファイルシステムテーブルに指定されているファイルシステムをマウントするときに使用します。ファイルシステムテーブルは vfstab(4) 形式である必要があります。*file_system_table* を指定しない場合は、/etc/vfstab が使用されます。*file_system_table* として – を指定すると、mountall は標準入力からファイルシステムテーブルを読み取ります。mountall は、*file_system_table* において mount at boot フィールドが yes に設定されているファイルシステムだけをマウントします。

ファイルシステムテーブル内のファイルシステムごとに、次のロジックが実行されます。 つまり、/usr/lib/fs/*FSType*/fsckall というファイル (*FSType* はファイルシステムのタイプ) が存在する場合、当該ファイルシステムをリストに保存しておき、後でまとめて /usr/lib/fs/*FSType*/fsckall スクリプトに引数として渡します。/usr/lib/fs/*FSType*/fsckall スクリプトは、引数リスト内にあるファイルシステムをすべて検査して、安全にマウントできるかどうかを決定します。*FSType* のファイルシステム用の /usr/lib/fs/*FSType*/fsckall スクリプトが存在しない場合、そのファイルシステムは fsck(1M) を使用して個々に検査されます。検査の結果、ファイルシステムがマウントできない状態であることが判明した場合、マウントを試行する前に fsck で修復されます。fsckdev フィールドのエントリが – であるファイルシステムは事前の検査なしにマウントされます。

umountall によって、root、/usr、/var、/var/adm、/var/run、/proc、および /dev/fd を除き、現在のゾーンにマウントされているすべてのファイルシステムがマウント解除されます。*FSType* を指定すると、mountall と umountall のアクションは指定された *FSType* に制限されます。-k を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。

オプション

次のオプションがサポートされています。

- **-F** マウントまたはマウント解除するファイルシステムの *FSType* を指定します。
- **-h *host*** host からリモートでマウントされている、/etc/mnttab 内のファイルシステムをすべてマウント解除します。
- **-k** fuser -k *mount-point* コマンドを使用します。詳細については、fuser(1M) を参照してください。-k オプションは SIGKILL シグナルをファイルを使用している各プロセスに送信します。このオプションはプロセスごとに終了シグナルを生成するので、終了メッセージがすぐに出力されないことがあります。-k を指定した場合でも、umountall がビジー状態のファイルシステムをマウント解除するかどうかは保証されません。
- **-l** アクションをローカルのファイルシステムに制限します。

-n　指定したオプションで実行されるアクションをリスト表示します。ただし、実際にはアクションを実行しません。-n オプションを指定せずにコマンドを繰り返すと、コマンドを繰り返す間に /etc/mnttab ファイルは変更されなかったと仮定して、リストされたアクションが実行されます。

-r　アクションをリモートのファイルシステムタイプに制限します。

-s　複数の umount 操作を同時に実行しないようにします。

-Z　非大域ゾーンにマウントされているファイルシステムにのみ操作を適用します。デフォルトでは、umountall は、現在のゾーンにマウントされているファイルシステムのみをマウント解除します。非大域ゾーンで使用する場合、オプション -Z は無視されます。

ファイル

/etc/mnttab
マウント済みファイルシステムテーブル

/etc/vfstab
デフォルトのファイルシステムテーブル

/usr/lib/fs/*FSType*/fsckall
FSType タイプのファイルシステムをすべて検査するときに mountall が呼び出すスクリプト

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実
出力の安定性	不確実

関連項目

fsck(1M), fuser(1M), mount(1M), mnttab(4), vfstab(4), attributes(5)

診断

ファイルシステムがマウント可能であり、クリーンな状態であれば、メッセージは出力されません。

エラーメッセージと警告メッセージは、fsck(1M) と mount(1M) の両方から送られます。

注意事項

この時点では、-l、-r、および -h オプションでサポートされるリモートファイルシステムは NFS のみです。

名前

mountd – server for NFS mount requests and NFS access checks

形式

```
/usr/lib/nfs/mountd [-v] [-r]
```

機能説明

mountd is an RPC server that answers requests for NFS access information and file system mount requests. It reads the file /etc/dfs/sharetab to determine which file systems are available for mounting by which remote machines. See sharetab(4). nfsd running on the local server will contact mountd the first time an NFS client tries to access the file system to determine whether the client should get read-write, read-only, or no access. This access can be dependent on the security mode used in the remoted procedure call from the client. See share_nfs(1M).

The command also provides information as to what file systems are mounted by which clients. This information can be printed using the showmount(1M) command.

The mountd daemon is automatically invoked by share(1M).

Only super user can run the mountd daemon.

SMF Management

Since mountd must be running for nfsd to function properly, mountd is automatically started by the svc:/network/nfs/server service.

Startup SMF parameters for mountd can be manipulated using the sharectl(1M) command. The currently supported parameters are:

client_versmin=*num*
client_versmax=*num*
: The NFS client uses only NFS versions in the range specified by these variables. Valid values or versions are: 2, 3, and 4. By default, these variables are unspecified (commented out) and the client's default minimum is Version 2. The default maximum is version 4. You can override this range on a per-mount basis by using the -o vers= option to mount_nfs(1M).

オプション

The options shown below are supported for NVSv2/v3 clients. They are not supported for Solaris NFSv4 clients.

-r
: Reject mount requests from clients. Clients that have file systems mounted will not be affected.

-v
: Run the command in verbose mode. Each time mountd determines what access a client should get, it will log the result to the console, as well as how it got that result.

ファイル

/etc/dfs/sharetab shared file system table

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

関連項目

mount_nfs(1M), nfsd(1M), share(1M), share_nfs(1M), sharectl(1M), showmount(1M), sharetab(4), attributes(5)

注意事項

Some routines that compare hostnames use case-sensitive string comparisons; some do not. If an incoming request fails, verify that the case of the hostname in the file to be parsed matches the case of the hostname called for, and attempt the request again.

名前

mount_hsfs – mount hsfs file systems

形式

```
mount -F hsfs [generic_options]
     [-o FSType-specific_options] [-O ] special | mount_point
```

```
mount -F hsfs [generic_options]
     [-o FSType-specific_options] [-O] special mount_point
```

機能説明

`mount` attaches an ISO 9660 filesystem (the High Sierra file system, `hsfs`, is a draft predecessor to ISO 9660, so the name reflects the filesystem's history) to the file system hierarchy at the *mount_point*, which is the pathname of a directory. If *mount_point* has any contents prior to the mount operation, these are hidden until the file system is unmounted.

If `mount` is invoked with *special* or *mount_point* as the only arguments, `mount` will search `/etc/vfstab` to fill in the missing arguments, including the *FSType-specific_options*; see `mount(1M)` for more details.

The `hsfs` file system supports direct mounting of files containing the file system as well as block devices. See `mount(1M)` and `lofiadm(1M)`.

A file system conforming to ISO 9660 can contain extensions that allow it to overcome limitations of the original ISO 9660:1988 (version 1) standard. The following types of extensions are supported by `hsfs`:

Rock Ridge extensions
: This is the preferred type of extension as it allows file attributes, name length, and types equivalent to those on other UNIX-style filesystems. Example of supported features are device special files, POSIX permissions, symbolic links, and filenames of up to 255 bytes in length. Rock Ridge extensions also remove the ISO9660:1988 restriction on maximum nesting depth for directories (eight levels). `hsfs` automatically detects the presence of Rock Ridge extensions and uses them, unless mount options are specified to disable the use of Rock Ridge or to use a different extension.

ISO9660:1999 (version 2) extensions
: The first version of ISO9660, released in 1988, supported only uppercase ASCII filenames of no more than 31 characters in length. ISO9660 version 2, released in 1999, provides an extension that allows filenames of at least 207 bytes that can use UTF-8 characters and removes the limitation on the nesting depth for directories. Unlike Rock Ridge, it does not provide support for UNIX-style file types and file attributes. `hsfs` automatically detects this extension and will use it for filename lookup if no Rock Ridge extensions are found on the media.

Joliet extensions
: The Joliet extension was devised by Microsoft to allow Unicode (UCS-2) long filenames with CDROM-based media. It allows filename lengths of up to 110 Unicode characters and does not support UNIX-style file types and attributes. `hsfs` falls back to using Joliet if such an extension is present and neither Rock Ridge nor ISO9660 version 2 extensions are found.

If filenames are longer than the 64 UCS-2 characters officially allowed by Microsoft (that is, 110 Unicode characters), they can translate to up to 330 UTF-8 octets. Filenames that translate to more than 255 UTF-8 octets will be truncated.

オプション

generic_options

See `mount(1M)` for the list of supported options.

`-o`

Specify `hsfs` file system specific options. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available:

`global | noglobal`

If `global` is specified and supported on the file system, and the system in question is part of a cluster, the file system will be globally visible on all nodes of the cluster. If `noglobal` is specified, the mount will not be globally visible. The default behavior is `noglobal`.

`ro`

Mount the file system read-only. This option is required.

`rr | nrr`

Enable (`rr`) or disable (`nrr`) the use of Rock Ridge. `rr` is the default and need not be specified. If you use `nrr` and Rock Ridge extensions are present in the file system, ignore them and search for other available extensions or fall back to plain ISO9660.

`vers2 | novers2`

Enable or disable the use of ISO9660 version 2 extensions. If `vers2` is specified and ISO9660 version 2 extensions are available, `hsfs` will use ISO9660 version 2 even if the file system contains the preferred Rock Ridge extensions as well. If `novers2` is specified, it will fall back to using either Joliet extensions or plain ISO9660 even if ISO9660 version 2 extensions are available.

`joliet | nojoliet`

Enable or disable the use of Joliet extensions. If `joliet` is specified and Joliet extensions are available, `hsfs` will use them even if the file system contains the preferred Rock Ridge and/or ISO9660 version 2 extensions. If `nojoliet` is specified, it will fall back to using plain ISO9660.

`notraildot`

File names on High Sierra file systems consist of a proper name and an extension separated by a '.' (dot) character. By default, the separating dot is always considered part of the file's name for all file access operations, even if there is no extension present. Specifying `notraildot` makes it optional to specify the trailing dot to access a file whose name lacks an extension.

Exceptions: This option is effective only on file systems for which Rock Ridge, ISO9660 version 2 or Joliet extensions are not active, either because they are not present on the CD-ROM, or they have been deliberately disabled via the `nrr`, `novers2` and `nojoliet` option. If either extension is active, `hsfs` quietly ignores this option.

`nomaplcase`
File names on High Sierra/ISO9660 CD–ROMs with no extensions present should be uppercase characters only. By default, `hsfs` maps file names read from a `non-Rock Ridge` disk to all lowercase characters. `nomaplcase` turns off this mapping. The exceptions for `notraildot` discussed above apply to `nomaplcase`.

`-O`
Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a preexisting mount point without setting this flag, the mount will fail, producing the error: `device busy`.

使用例

例 1 Mounting and Unmounting a DVD Image Directly

The following commands mount and unmount a DVD image.

```
# mount -F hsfs /images/solaris.iso /mnt/solaris-image
# umount /mnt/solaris-image
```

ファイル

`/etc/mnttab`
table of mounted file systems

`/etc/vfstab`
list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`lofiadm(1M)`, `mount(1M)`, `mountall(1M)`, `mount(2)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`

注意事項

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

名前

mount_nfs – mount remote NFS resources

形式

```
mount [-F nfs] [generic_options] [-o specific_options] [-O] resource

mount [-F nfs] [generic_options] [-o specific_options] [-O] mount_point

mount [-F nfs] [generic_options] [-o specific_options]
     [-O] resource mount_point
```

機能説明

The `mount` utility attaches a named *resource* to the file system hierarchy at the pathname location *mount_point*, which must already exist. If *mount_point* has any contents prior to the `mount` operation, the contents remain hidden until the *resource* is once again unmounted.

`mount_nfs` starts the `lockd(1M)` and `statd(1M)` daemons if they are not already running.

If the resource is listed in the `/etc/vfstab` file, the command line can specify either *resource* or *mount_point*, and `mount` consults `/etc/vfstab` for more information. If the `-F` option is omitted, `mount` takes the file system type from `/etc/vfstab`.

If the resource is not listed in the `/etc/vfstab` file, then the command line must specify both the *resource* and the *mount_point*.

host can be an IPv4 or IPv6 address string. As IPv6 addresses already contain colons, enclose *host* in a pair of square brackets when specifying an IPv6 address string. Otherwise the first occurrence of a colon can be interpreted as the separator between the host name and path, for example, `[1080::8:800:200C:417A]:tmp/file`. See `inet(7P)` and `inet6(7P)`.

host:*pathname*
: Where *host* is the name of the NFS server host, and *pathname* is the path name of the directory on the server being mounted. The path name is interpreted according to the server's path name parsing rules and is not necessarily slash-separated, though on most servers, this is the case.

nfs://*host*[:*port*]/*pathname*
: This is an NFS URL and follows the standard convention for NFS URLs as described in NFS URL Scheme, RFC 2224. See the discussion of URL's and the public option under `NFS FILE SYSTEMS` for a more detailed discussion.

host:*pathname* *nfs*://*host*[:*port*]/*pathname*
: *host*:*pathname* is a comma-separated list of *host*:*pathname*.

  See the discussion of replicated file systems and failover under `NFS FILE SYSTEMS` for a more detailed discussion.

hostlist pathname
: *hostlist* is a comma-separated list of hosts.

  See the discussion of replicated file systems and failover under `NFS FILE SYSTEMS` for a more detailed discussion.

The mount command maintains a table of mounted file systems in /etc/mnttab, described in mnttab(4).

mount_nfs supports both NFSv3 and NFSv4 mounts. The default NFS version is NFSv4.

SMF Management

The NFS client service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/nfs/client:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

オプション

See mount(1M) for the list of supported *generic_options*. See share_nfs(1M) for a description of server options.

-o *specific_options*
: Set file system specific options according to a comma-separated list with no intervening spaces.

acdirmax=*n*
: Hold cached attributes for no more than *n* seconds after directory update. The default value is 60.

acdirmin=*n*
: Hold cached attributes for at least *n* seconds after directory update. The default value is 30.

acregmax=*n*
: Hold cached attributes for no more than *n* seconds after file modification. The default value is 60.

acregmin=*n*
: Hold cached attributes for at least *n* seconds after file modification. The default value is 3.

actimeo=*n*
: Set *min* and *max* times for regular files and directories to *n* seconds. See “File Attributes,” below, for a description of the effect of setting this option to 0.

  See “Specifying Values for Attribute Cache Duration Options,” below, for a description of how acdirmax, acdirmin, acregmax, acregmin, and actimeo are parsed on a mount command line.

bg | fg
: If the first attempt fails, retry in the background, or, in the foreground. The default is fg.

`forcedirectio` | `noforcedirectio`
: If `forcedirectio` is specified, then for the duration of the mount, forced direct I/O is used. If the filesystem is mounted using `forcedirectio`, data is transferred directly between client and server, with no buffering on the client. If the filesystem is mounted using `noforcedirectio`, data is buffered on the client. `forcedirectio` is a performance option that is of benefit only in large sequential data transfers. The default behavior is `noforcedirectio`.

`grpid`
: By default, the GID associated with a newly created file obeys the System V semantics; that is, the GID is set to the effective GID of the calling process. This behavior can be overridden on a per-directory basis by setting the set-GID bit of the parent directory; in this case, the GID of a newly created file is set to the GID of the parent directory (see `open(2)` and `mkdir(2)`). Files created on file systems that are mounted with the `grpid` option obeys BSD semantics independent of whether the set-GID bit of the parent directory is set; that is, the GID is unconditionally inherited from that of the parent directory.

`hard` | `soft`
: Continue to retry requests until the server responds (`hard`) or give up and return an error (`soft`). The default value is `hard`. Note that NFSv4 clients do not support soft mounts.

`intr` | `nointr`
: Allow (do not allow) keyboard interrupts to kill a process that is hung while waiting for a response on a hard-mounted file system. The default is `intr`, which makes it possible for clients to interrupt applications that can be waiting for a remote mount.

`llock`
: Use local locking (no lock manager). Note that this is a private interface.

`noac`
: Suppress data and attribute caching. The data caching that is suppressed is the write-behind. The local page cache is still maintained, but data copied into it is immediately written to the server.

`nocto`
: Do not perform the normal close-to-open consistency. When a file is closed, all modified data associated with the file is flushed to the server and not held on the client. When a file is opened the client sends a request to the server to validate the client's local caches. This behavior ensures a file's consistency across multiple NFS clients. When `nocto` is in effect, the client does not perform the flush on close and the request for validation, allowing the possiblity of differences among copies of the same file as stored on multiple clients.

  This option can be used where it can be guaranteed that accesses to a specified file system are made from only one client and only that client. Under such a condition, the effect of `nocto` can be a slight performance gain.

`port=n`
: The server IP port number. The default is `NFS_PORT`. If the `port` option is specified, and if the resource includes one or more NFS URLs, and if any of the URLs include a `port` number, then the `port` number in the option and in the URL must be the same.

`posix`
: Request POSIX.1 semantics for the file system. Requires a mount Version 2 `mountd(1M)` on the server. See `standards(5)` for information regarding POSIX.

`proto=`*netid* | `rdma`
: By default, the transport protocol that the NFS mount uses is the first available RDMA transport supported both by the client and the server. If no RDMA transport is found, then it attempts to use a TCP transport or, failing that, a UDP transport, as ordered in the `/etc/netconfig` file. If it does not find a connection oriented transport, it uses the first available connectionless transport.

  Use this option to override the default behavior.

  `proto` is set to the value of *netid* or `rdma`. *netid* is the value of the `network_id` field entry in the `/etc/netconfig` file.

  The UDP protocol is not supported for NFS Version 4. If you specify a UDP protocol with the `proto` option, NFS version 4 is not used.

`public`
: The `public` option forces the use of the public file handle when connecting to the NFS server. The resource specified might not have an NFS URL. See the discussion of URLs and the public option under `NFS FILE SYSTEMS` for a more detailed discussion.

`quota` | `noquota`
: Enable or prevent `quota(1M)` to check whether the user is over quota on this file system; if the file system has quotas enabled on the server, quotas are still checked for operations on this file system.

`remount`
: Remounts a read-only file system as read-write (using the `rw` option). This option cannot be used with other `-o` options, and this option works only on currently mounted read-only file systems.

`retrans=n`
: Set the number of NFS retransmissions to *n*. The default value is `5`. For connection-oriented transports, this option has no effect because it is assumed that the transport performs retransmissions on behalf of NFS.

`retry=n`
: The number of times to retry the `mount` operation. The default for the `mount` command is `10000`.

The default for the automounter is `0`, in other words, do not retry. You might find it useful to increase this value on heavily loaded servers, where automounter traffic is dropped, causing unnecessary server not responding errors.

`rsize=`*n*
Set the read buffer size to a maximum of *n* bytes. The default value is `1048576` when using connection-orientated transports with Version 3 or Version 4 of the NFS protocol, and `32768` when using connection-less transports. The default can be negotiated down if the server prefers a smaller transfer size. "`Read`" operations may not necessarily use the maximum buffer size. When using Version 2, the default value is `32768` for all transports.

`sec=`*mode*
Set the security *mode* for NFS transactions. If `sec=` is not specified, then the default action is to use `AUTH_SYS` over NFS Version 2 mounts, use a user-configured default `auth` over NFS version 3 mounts, or to negotiate a mode over Version 4 mounts.

The preferred mode for NFS Version 3 mounts is the default mode specified in `/etc/nfssec.conf` (see `nfssec.conf(4)`) on the client. If there is no default configured in this file or if the server does not export using the client's default mode, then the client picks the first mode that it supports in the array of modes returned by the server. These alternatives are limited to the security flavors listed in `/etc/nfssec.conf`.

NFS Version 4 mounts negotiate a security mode when the server returns an array of security modes. The client attempts the mount with each security mode, in order, until one is successful.

Only one mode can be specified with the `sec=` option. See `nfssec(5)` for the available *mode* options.

`secure`
This option has been deprecated in favor of the `sec=`*dh* option.

`timeo=`*n*
Set the NFS timeout to *n* tenths of a second. The default value is `11` tenths of a second for connectionless transports, and `600` tenths of a second for connection-oriented transports. This value is ignored for connectionless transports. Such transports might implement their own timeouts, which are outside the control of NFS.

`vers=`*NFS version number*
By default, the version of NFS protocol used between the client and the server is the highest one available on both systems. The default maximum for the client is Version 4. This can be changed by setting `client_versmax` to a valid version number (`2`, `3`, or `4`). Use the `sharectl(1M)` command to manipulate the `client_versmax` property. If the NFS server does not support the client's default maximum, the next lowest version attempted until a matching version is found.

`wsize=n`
: Set the write buffer size to a maximum of *n* bytes. The default value is `1048576` when using connection-orientated transports with Version 3 or Version 4 of the NFS protocol, and `32768` when using connection-less transports. The default can be negotiated down if the server prefers a smaller transfer size. "`Write`" operations may not necessarily use the maximum buffer size. When using Version 2, the default value is `32768` for all transports.

`xattr|noxattr`
: Allow or disallow the creation and manipulation of extended attributes. The default is `xattr`. See `fsattr(5)` for a description of extended attributes.

`-O`
: Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount fails, producing the error "device busy."

Nfs File Systems

Background versus Foreground
: File systems mounted with the `bg` option indicate that `mount` is to retry in the background if the server's mount daemon (`mountd(1M)`) does not respond. `mount` retries the request up to the count specified in the `retry=n` option. (Note that the default value for `retry` differs between `mount` and `automount`. See the description of `retry`, above.) Once the file system is mounted, each NFS request made in the kernel waits `timeo=n` tenths of a second for a response. If no response arrives, the time-out is multiplied by `2` and the request is retransmitted. When the number of retransmissions has reached the number specified in the `retrans=n` option, a file system mounted with the `soft` option returns an error on the request; one mounted with the `hard` option prints a warning message and continues to retry the request.

Hard versus Soft
: File systems that are mounted read-write or that contain executable files should always be mounted with the `hard` option. Applications using `soft` mounted file systems can incur unexpected I/O errors, file corruption, and unexpected program core dumps. The soft option is not recommended.

Authenticated requests
: The server can require authenticated NFS requests from the client. `sec=dh` authentication might be required. See `nfssec(5)`.

URLs and the public option
: If the `public` option is specified, or if the *resource* includes and NFS URL, `mount` attempts to connect to the server using the public file handle lookup protocol. See WebNFS Client Specification, RFC 2054. If the server supports the public file handle, the attempt is successful; `mount` does not need to contact the server's `rpcbind(1M)` and the `mountd(1M)` daemons to get the port number of the `mount` server and the initial file handle of *pathname*, respectively. If the NFS client and server are separated by a firewall that allows all outbound connections through specific ports, such as `NFS_PORT`, then this enables NFS operations

through the firewall. The public option and the NFS URL can be specified independently or together. They interact as specified in the following matrix:

```
                  Resource Style

                   host:pathname              NFS URL

public option     Force public file          Force public file
                   handle and fail            handle and fail
                   mount if not supported.    mount if not supported.

                   Use Native paths.          Use Canonical paths.

default           Use MOUNT protocol.        Try public file handle
                                              with Canonical paths.
                                              Fall back to MOUNT
                                              protocol if not
                                              supported.
```

A Native path is a path name that is interpreted according to conventions used on the native operating system of the NFS server. A Canonical path is a path name that is interpreted according to the URL rules. See Uniform Resource Locators (URL), RFC 1738. See "Examples," below, for uses of Native and Canonical paths.

Replicated file systems and failover
resource can list multiple read–only file systems to be used to provide data. These file systems should contain equivalent directory structures and identical files. The file systems can be specified either with a comma–separated list of *host:/pathname* entries and/or NFS URL entries, or with a comma –separated list of hosts, if all file system names are the same. If multiple file systems are named and the first server in the list is down, failover uses the next alternate server to access files. If the read–only option is not chosen, replication is disabled. File access, for NFS Versions 2 and 3, is blocked on the original if NFS locks are active for that file.

File Attributes

To improve NFS read performance, files and file attributes are cached. File modification times get updated whenever a write occurs. However, file access times can be temporarily out-of-date until the cache gets refreshed.

The attribute cache retains file attributes on the client. Attributes for a file are assigned a time to be flushed. If the file is modified before the flush time, then the flush time is extended by the time since the last modification (under the assumption that files that changed recently are likely to change soon). There is a minimum and maximum flush time extension for regular files and for directories. Setting `actimeo=`*n* sets flush time to *n* seconds for both regular files and directories.

Setting `actimeo=0` disables attribute caching on the client. This means that every reference to attributes is satisfied directly from the server though file data is still cached. While this

guarantees that the client always has the latest file attributes from the server, it has an adverse effect on performance through additional latency, network load, and server load.

Setting the noac option also disables attribute caching, but has the further effect of disabling client write caching. While this guarantees that data written by an application is written directly to a server, where it can be viewed immediately by other clients, it has a significant adverse effect on client write performance. Data written into memory-mapped file pages (mmap(2)) are not written directly to this server.

Specifying Values for Attribute Cache Duration Options

The attribute cache duration options are acdirmax, acdirmin, acregmax, acregmin, and actimeo, as described under OPTIONS. A value specified for actimeo sets the values of all attribute cache duration options except for any of these options specified following actimeo on a mount command line. For example, consider the following command:

```
example# mount -o acdirmax=10,actimeo=1000 server:/path /localpath
```

Because actimeo is the last duration option in the command line, its value (1000) becomes the setting for all of the duration options, including acdirmax. Now consider:

```
example# mount -o actimeo=1000,acdirmax=10 server:/path /localpath
```

Because the acdirmax option follows actimeo on the command line, it is assigned the value specified (10). The remaining duration options are set to the value of actimeo (1000).

使用例

例 1 Mounting an NFS File System

To mount an NFS file system:

```
example# mount serv:/usr/src /usr/src
```

This is an example of the use of a native path.

例 2 Mounting An NFS File System Read-Only With No suid Privileges

To mount an NFS file system read-only with no suid privileges:

```
example# mount -r -o nosuid serv:/usr/src /usr/src
```

例 3 Mounting An NFS File System Over Version 2, with the UDP Transport

To mount an NFS file system over Version 2, with the UDP transport:

```
example# mount -o vers=2,proto=udp serv:/usr/src /usr/src
```

例 4 Mounting an NFS File System Using An NFS URL

To mount an NFS file system using an NFS URL (a canonical path):

```
example# mount nfs://serv/usr/man /usr/man
```

例 5 Mounting An NFS File System Forcing Use Of The Public File Handle

To mount an NFS file system and force the use of the public file handle and an NFS URL (a canonical path) that has a non 7–bit ASCII escape sequence:

```
example# mount -o public nfs://serv/usr/%A0abc /mnt/test
```

例 6 Mounting an NFS File System Using a Native Path

To mount an NFS file system using a native path (where the server uses colons (":") as the component separator) and the public file handle:

```
example# mount -o public serv:C:doc:new /usr/doc
```

例 7 Mounting a Replicated Set of NFS File Systems with the Same Pathnames

To mount a replicated set of NFS file systems with the same pathnames:

```
example# mount serv—a,serv—b,serv—c:/usr/man /usr/man
```

例 8 Mounting a Replicated Set of NFS File Systems with Different Pathnames

To mount a replicated set of NFS file systems with different pathnames:

```
example# mount serv—x:/usr/man,serv—y:/var/man,nfs://serv-z/man /usr/man
```

ファイル

`/etc/mnttab`
: table of mounted file systems

`/etc/dfs/fstypes`
: default distributed file system type

`/etc/vfstab`
: table of automatically mounted resources

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目

`lockd(1M)`, `mountall(1M)`, `mountd(1M)`, `nfsd(1M)`, `quota(1M)`, `sharectl(1M)`, `statd(1M)`, `mkdir(2)`, `mmap(2)`, `mount(2)`, `open(2)`, `umount(2)`, `mnttab(4)`, `nfssec.conf(4)`, `attributes(5)`, `fsattr(5)`, `nfssec(5)`, `standards(5)`, `inet(7P)`, `inet6(7P)`, `lofs(7FS)`

Callaghan, Brent, WebNFS Client Specification, RFC 2054, October 1996.

Callaghan, Brent, NFS URL Scheme, RFC 2224, October 1997.

Berners-Lee, Masinter & McCahill , Uniform Resource Locators (URL), RFC 1738, December 1994.

注意事項

An NFS server should not attempt to mount its own file systems. See `lofs(7FS)`.

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on *the directory to which the symbolic link refers,* rather than being mounted on top of the symbolic link itself.

SunOS 4.x used the `biod` maintenance procedure to perform parallel read-ahead and write-behind on NFS clients. SunOS 5.x made `biod` obsolete with multi-threaded processing, which transparently performs parallel read-ahead and write-behind.

Since the root `(/)` file system is mounted read-only by the kernel during the boot process, only the `remount` option (and options that can be used in conjunction with `remount`) affect the root `(/)` entry in the `/etc/vfstab` file.

名前

mount_pcfs – mount pcfs file systems

形式

```
mount -F pcfs [generic_options]
      [-o FSType-specific_options] special | mount_point

mount -F pcfs [generic_options]
      [-o FSType-specific_options] special mount_point
```

機能説明

`mount` attaches an MS-DOS file system (`pcfs`) to the file system hierarchy at the *mount_point*, which is the pathname of a directory. If *mount_point* has any contents prior to the `mount` operation, these are hidden until the file system is unmounted.

The `pcfs` file system supports direct mounting of files containing the file system as well as block devices. See `mount(1M)` and `lofiadm(1M)`.

If `mount` is invoked with *special* or *mount_point* as the only arguments, `mount` will search `/etc/vfstab` to fill in the missing arguments, including the *FSType-specific_options*; see `mount(1M)` for more details.

The *special* argument is a special device file type, which is:

A DOS logical drive on a hard disk expressed as *device-name*:*logical-drive* , where *device-name* specifies the special block device-file for the whole disk and *logical-drive* is either a drive letter (c through z) or a drive number (1 through 24). Examples are `/dev/dsk/c0t0d0p0:`*c* and `/dev/dsk/c0t0d0p0:`*1*.

The *special* device file type must have a formatted MS-DOS file system with either a 12-bit, 16-bit, or 32-bit File Allocation Table.

オプション

generic_options

See `mount(1M)` for the list of supported options.

`-o`

Specify `pcfs` file system-specific options. The following options are supported:

`clamptime` | `noclamptime`

File timestamps in `pcfs` cover a range between January 1st 1980 and December 31st 2127. This is not equal to the range of `time_t` on Unix for either 32-bit or 64-bit applications. In particular, 32-bit applications fail with `EOVERFLOW` errors on the `stat(2)` system call when timestamps beyond the range of 32-bit `time_t` are encountered. In order to prevent such spurious failures, `pcfs` by default clamps timestamps to the common subset of possible `pcfs` timestamps and the range available to 32-bit applications in Unix. The `clamptime` mount option therefore is active by default. If you want access to the full range of possible timestamps on `pcfs`, mount the file system with the `noclamptime` mount option. Note that if `noclamptime` is used, only 64-bit applications will have access to timestamps beyond January 19th 2038, 03:14:06 UTC; 32-bit applications will encounter `EOVERFLOW` errors.

`foldcase` | `nofoldcase`
: Force uppercase characters in filenames to lowercase when reading them from the filesystem. This is for compatibility with the previous behavior of `pcfs`. The default is `nofoldcase`.

`hidden` | `nohidden`
: Allow or disallow listing of files with hidden or system bits set. Option `hidden` is the default. When `nohidden` is effect, hidden and system files are neither visible nor accessible. Note that PCFS in previous releases of the Solaris operating system used the `nohidden` option as the default.

`atime` | `noatime`
: Enable or disable write access timestamps on DOS-formatted media. Default for fixed disks is `atime`, while for removable media `noatime` is used. The latter default is so that writes to flash-based media (“memory sticks”) can be minimized, to prolong lifetime.

`timezone=`*timezone*
: Timestamps on DOS-formatted media are recorded in the local time of the recording system. This can cause confusion when accessing removable media in which the recording and receiving system use different time zones. Use this option to force media timestamps to be interpreted for a specific time zone. The `mount_pcfs` command converts the given time zone name into a numerical offset that is passed to the `pcfs` kernel module, using the same rules as described in `environ(5)` for the `TZ` environment variable. By default, the `timezone` value is taken from the `TZ` environment variable.

`uid=`*user_id*
: Specifies the owner for the contents of the file system. *user_id* is a decimal value. The default owner is the user ID of current process.

`gid=`*group_id*
: Specifies the group ID for the contents of the file systems. *group_id* is a decimal value. The default group is the group ID of the current process.

`umask=`*user_mask*
: Specifies the mask used for setting maximum access permissions on contents of the file system. The *user_mask* is an octal value, a bitmask, where each bit represents the permissions that should be disabled. The default *user_mask* is the file mode creation mask of the mounting process.

  See `Intro(2)` for more information on masks.

ファイル

`/etc/mnttab`
: table of mounted file systems

`/etc/vfstab`
: list of default parameters for each file system

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/pcfs

関連項目

lofiadm(1M), mount(1M), mountall(1M), Intro(2), mount(2), stat(2), time(2), mnttab(4), vfstab(4), attributes(5), environ(5), pcfs(7FS)

注意事項

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

名前

mount_smbfs, umount_smbfs – 共有リソースを SMB ファイルサーバーからマウントおよびマウント解除する

形式

```
/usr/sbin/mount [-F smbfs] [generic-options] [-o name=value]
     [-O] resource

/usr/sbin/mount [-F smbfs] [generic-options] [-o name=value]
     [-O] mount-point

/usr/sbin/mount [-F smbfs] [generic-options] [-o name=value]
     [-O] resource mount-point

/usr/sbin/umount [-F smbfs] [generic-options] mount-point
```

機能説明

mount ユーティリティーは、指定されたリソース *resource* をファイルシステム階層内のパス名 *mount-point* の場所に接続します。このパス名は、すでに存在している場所でなければなりません。

mount 操作の実行前の時点で *mount-point* に内容が含まれる場合、これらの内容はリソースがマウント解除されるまで非表示のままになります。SYS_MOUNT 特権を持つ承認されたユーザーが mount 操作を実行できます。また、ユーザーは、自身の所有するディレクトリ上で SMBFS マウント操作を実行できます。

/etc/vfstab ファイルにリソースの一覧が記述されている場合、*resource* と *mount-point* のいずれかを指定できます。これは、mount コマンドが /etc/vfstab ファイルで詳細情報を参照するためです。-F オプションを省略すると、mount は /etc/vfstab ファイル内のエントリからファイルシステムのタイプを取得します。

リソースの一覧が /etc/vfstab ファイルに記述されていない場合は、コマンド行に *resource* と *mount-point* の両方を指定する必要があります。

umount ユーティリティーは、マウントしたファイルシステムをファイルシステム階層から切り離します。SYS_MOUNT 特権を持つ承認されたユーザーが umount 操作を実行できます。また、ユーザーは、自身の所有するディレクトリ上で SMBFS マウント解除操作を実行できます。

SMB 共有のマウントに成功するには、network/smb/client サービスが有効になっている必要があります。このサービスは、デフォルトで有効になっています。

サービスを有効にするには、次の svcadm(1M) コマンドを入力します。

```
# svcadm enable network/smb/client
```

オペランド

mount コマンドでは次のオペランドをサポートしています。

resource //*server*/*share*

マウントするリソースの名前。名前に加えて、リソースに関する次の情報も指定できます。

- *server* は、リモートコンピュータの DNS または NetBIOS 名です。
- *share* は、リモートサーバー上のリソース名です。

ユーザーアカウントも指定できます。「オプション」の節を参照してください。

`mount` コマンドは、ユーザーアカウントの標準入力からパスワードを読み取ることができます。パスワードが指定されない場合、`mount` は最初に、`smbadm add-key` コマンドによって格納されたパスワード (存在する場合) の使用を試みます。そのパスワードが認証に失敗した場合、標準入力が TTY であれば、`mount_smbfs` コマンドがパスワードの入力をユーザーに求めます。

mount-point

ファイルシステムをマウントまたはマウント解除する場所へのパス。`mount` コマンドは、マウントしたファイルシステムのテーブルを `/etc/mnttab` ファイル内で管理します。`mnttab(4)` のマニュアルページを参照してください。

オプション

サポートされる *generic-options* のリストについては、`mount(1M)` のマニュアルページを参照してください。

`-o` *name*=*value* または
`-o` *name*

ファイルシステム固有のプロパティーを設定します。複数の名前-値ペアを、コンマ区切りのリストで指定できます。リストにスペースを含めることはできません。プロパティーは次のとおりです。

`acdirmax=`*n*

キャッシュした属性を保持する時間は、ディレクトリの更新後 *n* 秒以内です。デフォルト値は `60` です。

`acdirmin=`*n*

キャッシュした属性を保持する時間は、ディレクトリの更新後 *n* 秒以上です。デフォルト値は `30` です。

`acregmax=`*n*

キャッシュした属性を保持する時間は、ファイルの変更後 *n* 秒以内です。デフォルト値は `60` です。

`acregmin=`*n*

キャッシュした属性を保持する時間は、ファイルの変更後 *n* 秒以上です。デフォルト値は `3` です。

`actimeo=`*n*

通常のファイルおよびディレクトリの最小と最大の時間値を *n* 秒に設定します。このオプションを `0` に設定した場合の効果については、後述の「ファイル属性」を参照してください。

mount コマンド行で acdirmax、acdirmin、acregmax、acregmin、および actimeo が解析される方法については、後述の「属性キャッシュ期間オプションの値の指定」を参照してください。

dirperms=*octaltriplet*
: ディレクトリに割り当てるアクセス権を設定します。値は 3 つの 8 進数で指定する必要があります (例: 755)。ディレクトリモードのデフォルト値は fileperms 設定から取得され、fileperms が読み取り権を持つ場所に実行権が追加されます。

  これらのアクセス権は、SMB サーバーにより許可される権限とは無関係です。

domain=value
: ユーザー名が定義されているワークグループまたは Windows ドメインの名前を指定します。ドメイン名を指定しない場合、デフォルトのシステムの SMB ドメインが使用されます。

fileperms=*octaltriplet*
: ファイルに割り当てるアクセス権を設定します。値は 3 つの 8 進数で指定する必要があります (例: 644)。デフォルト値は 700 です。

  これらのアクセス権は、SMB サーバーにより許可される権限とは無関係です。

gid=*groupid*
: 指定したグループ ID をファイルに割り当てます。デフォルト値は、ボリュームがマウントされているディレクトリのグループ ID です。

intr|nointr
: ユーザーが呼び出し元のスレッドに割り込みを行なった場合 (たとえば、操作の実行中に Ctrl-C キーを押した場合) に、smbfs(7FS) I/O 操作の取り消しを有効 (または無効) にします。デフォルトは intr (割り込み有効) で、通常は取り消しが許可されます。

noac
: 属性のキャッシュを抑制します。ローカルの stat(2) 呼び出しでは、常に SMB サーバーに属性を要求します。

noprompt
: 共有のマウント時でのパスワードの入力要求を抑制します。このプロパティーを指定すると、共有への匿名アクセスが可能になります。匿名アクセスでは、パスワードは不要です。

  mount 操作が失敗するのは、パスワードが必須で、noprompt プロパティーが設定されており、smbadm add-key コマンドによって格納されたパスワードがない場合です。

`retry_count=`*number*
接続に切断のマークが付けられるまでの SMBFS の再試行回数を指定します。デフォルトの試行回数は 4 回です。

`timeout=`*seconds*
SMB 要求のタイムアウトを指定します。デフォルトでは、タイムアウトは 15 秒です。

`uid=`*userid*
指定したユーザー ID をファイルに割り当てます。デフォルト値は、ボリュームがマウントされているディレクトリの所有者 ID です。

user=value
リモートユーザー名を指定します。ユーザーを省略した場合、ログインしているユーザー ID が使用されます。

`xattr|noxattr`
このマウントポイントで Solaris 拡張属性を有効 (または無効) にします。このオプションのデフォルトは `xattr` (拡張属性の有効化) ですが、SMB サーバーが SMB の「名前付きストリーム」に対応していない場合、`smbfs(7FS)` はこのオプションを強制的に `noxattr` に設定します。mount コマンドに `noxattr` オプションが指定されている場合、Solaris 拡張属性の使用は EINVAL で失敗します。

`-O`
マウントをオーバーレイします。既存のマウントポイント上にファイルシステムをマウントできます。これより基礎となるファイルシステムにはアクセスできなくなります。このフラグを設定せずに既存のマウントポイント上にファイルシステムをマウントしようとすると、マウントは失敗して、device busy というエラーメッセージが表示されます。

ファイル属性

`smbfs` のパフォーマンスを改善するために、ファイル属性がキャッシュされます。ローカルで変更が行われると、常にファイルの変更時間が更新されます。ただし、ファイルのアクセス時間は、キャッシュが更新されるまで一時的に古いままになることがあります。

属性のキャッシュには、クライアント上のファイル属性が保持されます。ファイルの属性には、フラッシュされる時間が割り当てられます。フラッシュの時間より前にファイルが変更された場合、(最近変更されたファイルはまもなく変更されるだろうという仮定に基づき) 最終変更以降の時間だけフラッシュの時間が延長されます。通常ファイルおよびディレクトリの場合、フラッシュの延長時間には最小値と最大値があります。`actimeo=`*n* を設定すると、通常ファイルとディレクトリの両方で、フラッシュ時間が *n* 秒に設定されます。

`actimeo=`*n* を設定すると、クライアントでの属性のキャッシュが無効になります。これは、属性を参照すると、そのすべての情報がサーバーから直接送られることを意味します。これにより、クライアントが常にサーバーから最新のファイル属

性を入手することが保証されますが、待ち時間、ネットワークの負荷、およびサーバーの負荷が増加するため、パフォーマンスが悪影響を受けます。

noac オプションを設定しても、属性のキャッシュが無効になります。smbfs を拡張して書き込みキャッシュをサポートしている場合、このオプションにより書き込みキャッシュも無効になります。

属性キャッシュ期間オプションの値の指定

前述の「オプション」で説明したように、属性キャッシュ期間オプションには acdirmax、acdirmin、acregmax、acregmin、および actimeo があります。actimeo に設定した値は、mount コマンド行の actimeo のあとに指定されたオプションを除き、すべての属性キャッシュ期間オプションに適用されます。たとえば、次のコマンドについて考えてみます。

```
# mount -F smbfs -o acdirmax=10,actimeo=1000 \
     //server/share  /mountpoint
```

actimeo はコマンド行の末尾に記述された期間オプションであるため、その値 (1000) が acdirmax を含む、すべての期間オプションの設定になります。次について考えてみます。

```
# mount -F smbfs -o actimeo=1000,acdirmax=10 \
     //server/share  /mountpoint
```

acdirmax オプションはコマンド行で actimeo のあとに配置されているため、このオプションには指定された値 (10) が割り当てられます。残りの期間オプションには、actimeo (1000) の値が設定されます。

使用例

例 1　SMBFS 共有をマウントする

次の例では、SALES ワークグループの nano サーバー上にある /tmp 共有を、ローカルの /mnt マウントポイントにマウントする方法を示します。マウント操作に成功するには、root ユーザーのパスワードを指定する必要があります。

```
# mount -F smbfs -o user=root,domain=SALES //nano.sfbay/tmp /mnt
Password:
```

例 2　SMBFS ファイルシステムがマウントされていることを確認する

次の例では、nano サーバーの /tmp 共有をローカルの /mnt マウントポイントにマウントする方法を示します。マウント操作に成功するには、root ユーザーのパスワードを指定する必要があります。

```
# mount -F smbfs -o user=root //nano.sfbay/tmp /mnt
Password:
```

次の方法で、共有がマウントされていることを確認できます。

- /etc/mnttab ファイルでファイルシステムのエントリを参照します。

```
# grep mnt /etc/mnttab
//nano.sfbay/tmp   /mnt    smbfs   dev=4900000     1177097833
```

例 2 SMBFS ファイルシステムがマウントされていることを確認する (続き)

- mount コマンドの出力を参照します。

```
# mount | grep mnt
mnt on //nano.sfbay/tmp read/write/setuid/devices/dev=4900000 on
Tue Apr 20 13:37:13 2010
```

- df /mnt コマンドの出力を参照します。

```
# df /mnt
/mnt            (//nano.sfbay/tmp): 3635872 blocks     -1 files
```

df -k /mnt コマンドの出力を参照して、マウント済みの共有に関する情報を取得します。

```
# df -k /mnt
Filesystem            kbytes    used   avail capacity  Mounted on
//nano.sfbay/tmp
                      1882384   64448 1817936     4%    /mnt
```

例 3 SMB 共有をマウント解除する

この例では、SMB 共有が /mnt マウントポイントにマウントされているものとします。次のコマンド行により、共有がマウントポイントからマウント解除されます。

```
# umount /mnt
```

ファイル

/etc/mnttab
マウントされているファイルシステムのテーブル

/etc/dfs/fstypes
デフォルトの分散ファイルシステムタイプ。

/etc/vfstab
自動的にマウントされたリソースのテーブル。

/var/smb/smbfspasswd
Solaris SMB クライアントのユーザーごとの設定を格納します。

属性

次の属性についての詳細は、attributes(5) のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	system/file-system/smb
インタフェースの安定性	確実

関連項目

mount(1M), mountall(1M), smbadm(1M), svcadm(1M), acl(2), fcntl(2), link(2), mknod(2), mount(2), stat(2), symlink(2), umount(2), mnttab(4), smb(4), vfstab(4), attributes(5), pcfs(7FS), smbfs(7FS)

注意事項

Solaris SMB クライアントは、gethostbyname() を使用してホスト名を解決することを常に試みます。ホスト名を解決できない場合、SMB クライアントは NetBIOS 名前解決 (NBNS) を使用します。Solaris SMB クライアントでは、Windows 環境の Solaris SMB クライアントが追加の設定なしで動作できるように NBNS の使用を許可しています。

ファイルシステムがマウントされるディレクトリがシンボリックリンクの場合、ファイルシステムは、シンボリックリンク自身ではなく、シンボリックリンクが参照するディレクトリ上にマウントされます。

名前

mount_tmpfs – mount tmpfs file systems

形式

`mount [-F tmpfs] [-o` *specific_options*`] [-O]` *special mount_point*

機能説明

`tmpfs` is a memory based file system which uses kernel resources relating to the VM system and page cache as a file system.

`mount` attaches a `tmpfs` file system to the file system hierarchy at the pathname location *mount_point*, which must already exist. If *mount_point* has any contents prior to the `mount` operation, these remain hidden until the file system is once again unmounted. The attributes (mode, owner, and group) of the root of the `tmpfs` filesystem are inherited from the underlying *mount_point*, provided that those attributes are determinable. If not, the root's attributes are set to their default values.

The *special* argument is usually specified as `swap` but is in fact disregarded and assumed to be the virtual memory resources within the system.

オプション

`-o` *specific_options*

Specify `tmpfs` file system specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available:

size=*sz*

The *sz* argument controls the size of this particular `tmpfs` file system. If the argument is has a 'k' suffix, the number will be interpreted as a number of kilobytes. An 'm' suffix will be interpreted as a number of megabytes. No suffix is interpreted as bytes. In all cases, the actual size of the file system is the number of bytes specified, rounded up to the physical pagesize of the system.

`xattr` | `noxattr`

Allow or disallow the creation and manipulation of extended attributes. The default is `xattr`. See `fsattr(5)` for a description of extended attributes.

`-O`

Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount will fail, producing the error`device busy`.

ファイル

`/etc/mnttab` Table of mounted file systems

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

mount(1M), mkdir(2), mount(2), open(2), umount(2), mnttab(4), attributes(5), fsattr(5), tmpfs(7FS)

注意事項

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

名前

mount_udfs – mount a udfs file system

形式

```
mount -F udfs [generic_options] [-o specific_options]
      [-O] special mount_point
```

```
mount -F udfs [generic_options] [-o specific_options]
      [-O] special | mount_point
```

機能説明

The `mount` utility attaches a `udfs` file system to the file system hierarchy at the *mount_point*, which is the pathname of a directory. If *mount_point* has any contents prior to the mount operation, these are hidden until the file system is unmounted.

If mount is invoked with either *special* or *mount_point* as the only arguments, `mount` searches `/etc/vfstab` to fill in the missing arguments, including the *specific_options*. See `mount(1M)`.

The `udfs` file system supports direct mounting of files containing the file system as well as block devices. See `mount(1M)` and `lofiadm(1M)`.

If *special* and *mount_point* are specified without any *specific_options*, the default is `rw`.

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

オプション

See `mount`(1M) for the list of supported *generic_options*.

The following options are supported:

`-o` *specific_options*

Specify `udfs` file system specific options in a comma-separated list with no intervening spaces. The following *specific_options* are available:

m

Mount the file system without making an entry in `/etc/mnttab`.

remount

Remount the file system as read-write. The option is used in conjunction with the `rw` option.

A file system mounted read-only can be remounted as read-write. This option fails if the file system is not currently mounted.

`-O`

Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, the mount fails, producing the error `device busy`.

ファイル

`/etc/mnttab`
: Table of mounted file systems

`/etc/vfstab`
: List of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/udfs

関連項目

`fsck(1M)`, `fsck_udfs(1M)`, `lofiadm(1M)`, `mount(1M)`, `mountall(1M)`, `mount(2)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`

診断

not super user
: The command is run by a non-root user. Run as root.

no such device
: The device name specified does not exist.

not a directory
: The specified mount point is not a directory.

is not an udfs file system
: The device specified does not contain a udf 1.50 file system or the udfs file system module is not available.

is already mounted
: The specified device is already in use.

not a block device
: The device specified is not a block device. Use block device to mount.

write-protected
: The device is read-only.

is corrupted. needs checking
: The file system is in an inconsistent state. Run `fsck`.

注意事項

Copy-protected files can be stored on DVD-ROM media using Universal Disk Format (UDF). Reading these copy-protected files is not possible as this involves an authentication process. Unless an authentication process between the host and the drive is completed, reading these copy-protected files after mounting and before the authentication process, returns an error.

名前

mount_ufs – mount ufs file systems

形式

```
mount -F ufs [generic_options] [-o specific_options]
     [-O] special | mount_point
```

```
mount -F ufs [generic_options] [-o specific_options]
     [-O] special mount_point
```

機能説明

The `mount` utility attaches a `ufs` file system to the file system hierarchy at the *mount_point*, which is the pathname of a directory. If *mount_point* has any contents prior to the `mount` operation, these are hidden until the file system is unmounted.

The `ufs` file system supports direct mounting of files containing the file system as well as block devices. See `mount(1M)` and `lofiadm(1M)`.

If `mount` is invoked with *special* or *mount_point* as the only arguments, `mount` will search `/etc/vfstab` to fill in the missing arguments, including the *specific_options*. See `mount(1M)`.

If *special* and *mount_point* are specified without any *specific_options*, the default is `rw`.

If the directory on which a file system is to be mounted is a symbolic link, the file system is mounted on the directory to which the symbolic link refers, rather than on top of the symbolic link itself.

オプション

See `mount(1M)` for the list of supported *generic_options*.

The following options are supported:

`-o` *specific_options*

Specify `ufs` file system specific options in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is printed and the invalid options are ignored. The following options are available:

`dfratime` | `nodfratime`

By default, writing access time updates to the disk may be deferred (`dfratime`) for the file system until the disk is accessed for a reason other than updating access times. `nodfratime` disables this behavior.

If power management is enabled on the system, do not set `nodfratime` unless `noatime` is also set. If you set `nodfratime` without setting `noatime`, the disk is spun up every time a file within a file system on the disk is accessed - even if the file is not modified.

`forcedirectio` | `noforcedirectio`

If `forcedirectio` is specified and supported by the file system, then for the duration of the mount, forced direct I/O will be used. If the filesystem is mounted using `forcedirectio`, data is transferred directly between user address space and the disk. If the filesystem is mounted using `noforcedirectio`, data is buffered in kernel address space when data is transferred between user address space and the disk. `forcedirectio` is a performance option that is of benefit only in large sequential data transfers. The default behavior is `noforcedirectio`.

`global` | `noglobal`
: If `global` is specified and supported on the file system, and the system in question is part of a cluster, the file system will be globally visible on all nodes of the cluster. If `noglobal` is specified, the mount will not be globally visible. The default behavior is `noglobal`.

`intr` | `nointr`
: Allow (do not allow) keyboard interrupts to kill a process that is waiting for an operation on a locked file system. The default is `intr`.

`largefiles` | `nolargefiles`
: If `nolargefiles` is specified and supported by the file system, then for the duration of the mount it is guaranteed that all regular files in the file system have a size that will fit in the smallest object of type `off_t` supported by the system performing the mount. The mount will fail if there are any files in the file system not meeting this criterion. If `largefiles` is specified, there is no such guarantee. The default behavior is `largefiles`.

  If `nolargefiles` is specified, `mount` will fail for `ufs` if the file system to be mounted has contained a large file (a file whose size is greater than or equal to 2 Gbyte) since the last invocation of `fsck` on the file system. The large file need not be present in the file system at the time of the mount for the mount to fail; it could have been created previously and destroyed. Invoking `fsck` (see `fsck_ufs(1M)`) on the file system will reset the file system state if no large files are present. After invoking `fsck`, a successful mount of the file system with `nolargefiles` specified indicates the absence of large files in the file system; an unsuccessful mount attempt indicates the presence of at least one large file.

`logging` | `nologging`
: If `logging` is specified, then logging is enabled for the duration of the mounted file system. Logging is the process of storing transactions (changes that make up a complete UFS operation) in a log before the transactions are applied to the file system. Once a transaction is stored, the transaction can be applied to the file system later. This prevents file systems from becoming inconsistent, therefore reducing the possibility that `fsck` might run. And, if `fsck` is bypassed, logging generally reduces the time required to reboot a system.

  The default behavior is `logging` for all UFS file systems.

  The log is allocated from free blocks in the file system, and is sized approximately 1 Mbyte per 1 Gbyte of file system, up to a maximum of 256 Mbytes. The log size may be larger (up to a maximum of 512 Mbytes) dependent upon the number of cylinder groups present in the file system.

  Logging is enabled on any UFS file system, including root (`/`), except under the following conditions:

  - When logging is specifically disabled.
  - If there is insufficient file system space for the log. In this case, the following message is displayed and file system is still mounted:

```
# mount /dev/dsk/c0t4d0s0 /mnt
  /mnt: No space left on device
  Could not enable logging for /mnt on  /dev/dsk/c0t4d0s0.
```

The log created by UFS logging is continually flushed as it fills up. The log is totally flushed when the file system is unmounted or as a result of the `lockfs -f` command.

`m`
Mount the file system without making an entry in `/etc/mnttab`.

`noatime`
By default, the file system is mounted with normal access time (`atime`) recording. If `noatime` is specified, the file system will ignore access time updates on files, except when they coincide with updates to the `ctime` or `mtime`. See `stat(2)`. This option reduces disk activity on file systems where access times are unimportant (for example, a Usenet news spool).

`noatime` turns off access time recording regardless of `dfratime` or `nodfratime`.

The POSIX standard requires that access times be marked on files. `-noatime` ignores them unless the file is also modified.

`nosec`
By default, Access Control Lists (ACLs) are supported on a mounted UFS file system. Use this option to disallow the setting or any modification of an ACL on a file within a mounted UFS file system. See `getfacl(1)` for background on ACLs.

`onerror` = *action*
This option specifies the action that UFS should take to recover from an internal inconsistency on a file system. Specify *action* as `panic`, `lock`, or `umount`. These values cause a forced system shutdown, a file system lock to be applied to the file system, or the file system to be forcibly unmounted, respectively. The default is `panic`.

`quota`
Quotas are turned on for the file system.

`remount`
Remounts a file system with a new set of options. All options not explicitly set with `remount` revert to their default values.

`rq`
Read-write with quotas turned on. Equivalent to `rw, quota`.

`xattr` | `noxattr`
Allow or disallow the creation and manipulation of extended attributes. The default is `xattr`. See `fsattr(5)` for a description of extended attributes.

`-O`
Overlay mount. Allow the file system to be mounted over an existing mount point, making the underlying file system inaccessible. If a mount is attempted on a pre-existing mount

point without setting this flag, the mount will fail, producing the error “device busy”.

使用例

例 1 Turning Off (and On) Logging

The following command turns off logging on an already mounted file system. The subsequent command restores logging.

```
# mount -F ufs -o remount,nologging /export
# (absence of message indicates success)
# mount -F ufs -o remount,logging /export
```

In the preceding commands, the `-F ufs` option is not necessary.

ファイル

`/etc/mnttab`
table of mounted file systems

`/etc/vfstab`
list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`getfacl(1)`, `fsck(1M)`, `fsck_ufs(1M)`, `lofiadm(1M)`, `mount(1M)`, `mountall(1M)`, `fcntl(2)`, `mount(2)`, `stat(2)`, `mnttab(4)`, `vfstab(4)`, `attributes(5)`, `fsattr(5)`, `largefile(5)`

注意事項

Since the root (`/`) file system is mounted read-only by the kernel during the boot process, only the `remount` option (and options that can be used in conjunction with `remount`) affect the root (`/`) entry in the `/etc/vfstab` file.

名前

mpathadm – multipath discovery and administration

形式

`mpathadm` *subcommand direct-object* [*options*] [*operand*]

機能説明

The `mpathadm` command enables multipathing discovery and management. The `mpathadm` command is implemented as a set of subcommands, many with their own options, that are described in the section for that subcommand. Options not associated with a particular subcommand are described under OPTIONS. The `mpathadm` subcommands operate on a *direct-object*. These are described in this section for each subcommand. The *direct-objects*, *initiator-port*, *target-port*, and*logical-unit* in the subcommands are consistent with SCSI standard definitions.

The `mpathadm` command supports the following subcommands, which are described in detail in subsections that follow.

`list`	Display a list of discovered instances for a given object.
`show`	Display information about a given object instance.
`modify`	Modify properties of an object.
`enable`	Enable an object.
`disable`	Disable an object.
`failover`	Cause target port group failover for a logical-unit.
`override`	Set a path to be used over other paths on a logical-unit.

The `mpathadm` subcommands operate on a *direct-object*. These are described in this section for each subcommand.

list Subcommand

The syntax for the `list` subcommand is:

```
# mpathadm list direct-object [operands...]
```

The `list` subcommand displays data for following direct-objects:

`mpath-support` [*mpath-support-name*, ...]
: List the multipathing support that can be administered by this CLI. This presents itself in the form of a library name registered through the MPAPI framework. If no mpath-support name *mpath-support-name* is specified, all registered multipathing support libraries will be displayed.

`initiator-port` [*initiator-port-name*, ...]
: List the initiator ports that are discovered on this system. If no *initiator-port-name* is specified, all discovered initiator ports are displayed.

{`logical-unit` | `lu`} [*options*] [*logical-unit-name*, ...]
: List the information on multipath logical units. If no *logical-unit-name* is specified, all discovered logical-units will be displayed.

Options for `list logical-unit` are as follows:

`-n`, `--name` *name*	Return the logical unit name that is associated with the given name string. This name can be extracted from the output of the `mpathadm show lu` command.
`-t`, `--target-port` *target-port-name*	Return the list of logical units names that are associated with the given *target-port-name*.

show Subcommand

The syntax for the `show` subcommand is:

```
# mpathadm show direct-object [operands...]
```

The `show` subcommand displays detailed information for following the direct-objects:

`mpath-support` [*mpath-support-name*, ...]
: Show the detailed information on the given *mpath-support-name* if the name exists. If the given *mpath-support-name* supports only a limited set of device products, the list of device products will be listed as part of the output.

`initiator-port` *initiator-port-name*[,*initiator-port-name*, ...]
: Show the detailed information for the given *initiator-port-name*.

{`logical-unit` | `lu`} [*logical-unit-name*, ...]
: Display the detailed information on multipath logical unit(s), including path and target port group information. Note that the name property in the logical unit information represents the identifier for this LUN, derived from the hardware, and used by this system. If the name is derived from SCSI Inquiry Vital Product Data (VPD) page 83h, the name type property represents an associated identifier type defined by the SCSI standards.

modify Subcommand

The syntax for the `modify` subcommand is:

```
# mpathadm modify direct-object [options] [operands...]
```

The `modify` subcommand modifies characteristics of the following direct-objects:

`mpath-support` [*options*] *mpath-support-name*, ...	Configuration management of an *mpath-support*. Options to modify mpath-support are as follows: Set `autofailback` to `on`	`off`. Applicable only when mpath-support provides auto failback support. Set `autoprobing` to `on`	`off`. Applicable only when mpath-support provides auto probing support. Change the load balance type. The `loadbalance` type is one of the

supported types listed in the `show mpath-support` output.

{`logical-unit` | `lu`} [*options*] *logical-unit-name*, ...

Configuration management of a logical unit. Options to modify `logical-unit` are as follows:

`-a` *autofailback* Set auto failback. Applicable only when `mpath-support` provides auto failback support

`-p` *autoprobing* Set auto probing. Applicable only when `mpath-support` provides auto probing support.

`-b` *load-balance-type* Set load balance type for the logical unit. Applicable only when load balance configuration is supported at the logical unit level.

enable Subcommand

The syntax for the `enable` subcommand is:

```
# mpathadm disable [options]
```

The `enable` subcommand supports the following direct-objects to be enabled:

`path -i` *initiator-port-name* `-t` *target-port-name* `-l` *logical-unit-name*

The path that consists of the specified initiator port, target port, and logical unit will be enabled.

disable Subcommand

The syntax for the `disable` subcommand is:

```
# mpathadm disable [options]
```

The `disable` subcommand supports the following direct-objects to be disabled:

`path -i` *initiator-port-name* `-t` *target-port-name* `-l` *logical-unit-name*

The path that consists of the specified initiator port, target port, and logical unit will be disabled.

failover Subcommand

The syntax for the `failover` subcommand is:

```
# mpathadm failover direct-object [operand]
```

The `failover` subcommand supports failover for the following direct-objects:

{`logical-unit` | `lu`} *logical-unit-name*

The target port group will failover when the given logical-unit is asymmetric and supports explicit

state change. The currently active target port group will be changed to the standby state and the standby target port group will be active.

override Subcommand

The syntax for the `override` subcommand is:

```
# mpathadm override [options]
```

The `override` subcommand controls whether or not the following direct-objects override another:

path {`-i` *initiator-port-name* `-t` *target-port-name* | `-c`}
`-l` *logical-unit-name*

Cause a path that consists of the specified initiator port, target port, and logical unit to override other paths on the logical unit. Once a path overrides other paths, the mpath-support uses only that path regardless of any other path selection configuration. The `-c` option cancels the setting. The path that consists of the specified initiator port, target port, and logical unit will be disabled.

Options for override path are as follows:

`-i`, `--initiator-port` *initiator-port-name*	Represent the initiator port element of a path. Options `-t` and `-l` must also be included.
`-t`, `--target-port` *target-port-name*	Represent the target port element of a path. Options `-i` and `-l` must also be included.
`-l`, `--logical-unit` *logical-unit*	Represent the logical unit element of a path. Options `-i` and `-t` must also be included.
`-c`, `--cancel`	Cancels overriding setting for the given logical unit. Option `-l` must also be included.

オプション

The following options are supported:

`-V`, `--version`	Display the version information.
`-?`, `--help`	Display context help. Can be used following an `mpathadm` command with no arguments, following a subcommand, or following a subcommand direct-object combination. Responds with help information appropriate for your entry. For example, if you enter: **`# mpathadm add mpath-support-help`** ... `mpathadm` responds with a display of the options available for that combination of subcommand and direct-object.

使用例

例 1 Obtaining a List of Multipathing Support

The following command uses the `list` subcommand to list all currently registered `mpath-support` libraries.

```
# mpathadm list mpath-support
mpath-support: libmpscsi_vhci.so
```

例 2 Displaying the Properties of a Particular Multipathing Support

The following command uses the `show` subcommand to display the properties for a currently registered `mpath-support` library.

```
# mpathadm show mpath-support libmpscsi_vhci.so
mpath-support:    libmpscsi_vhci.so
       Vendor:  Sun Microsystems
       Driver Name:  scsi_vhci
       Default Load Balance:  round-robin
       Supported Load Balance Types:
              round-robin
          logical-block
Allows To Activate Target Port Group Access:  yes
Allows Path Override:  no
Supported Auto Failback Config:     1
Auto Failback:    on
Failback Polling Rate (current/max):  0/0
Supported Auto Probing Config:    0
Auto Probing:  NA
Probing Polling Rate (current/max):  NA/NA
Supported Devices:
       Vendor:     SUN
       Product:  T300
       Revision:
       Supported Load Balance Types:
              round-robin
       Vendor:     SUN
       Product:  T4
       Revision:
       Supported Load Balance Types:
              round-robin
```

例 3 Obtaining a List of Initiator Ports Discovered Through the mpath-support Libraries

The following command uses the `list initiator-port` subcommand to display a list of initiator ports discovered by the currently registered `mpath-support` libraries.

```
# mpathadm list initiator-port
Initiator-Port: iqn.1986-03.com.sun:01:080020b7ac2b.437a3b3e,4000002a0000
Initiator-Port: 2000000173018713
Initiator-Port: 2000000173818713
```

例 4 Displaying the Properties of a Particular Initiator Port

The following command uses the `show initiator-port` subcommand to display the properties of a particular initiator port discovered using the `list initiator-port` subcommand in an example above.

```
# mpathadm show initiator-port 2000000173018713
initiator-port:     2000000173018713
        Transport Type:     Fibre Channel
        OS device File:     devices/pci@1f,4000/pci@2/SUNW,qlca@5/fp@0,0:fc
```

例 5 Displaying the Properties of a Particular Logical Unit

The following command uses the `show logical-unit` subcommand to display the properties of the logical unit with the specified name.

```
# mpathadm show lu /dev/rdsk/c4t60003BA27D2120004204AC2B000DAB00d0s2
Logical Unit: /dev/rdsk/c4t60003BA27D2120004204AC2B000DAB00d0s2
           mpath-support libmpscsi_vhci.so
           Vendor: SUN
           Product: T4
           Revision: 0301
    Name Type: SCSI Inquiry VPD Page 83 type 3
    Name: 60003ba27d2120004204ac2b000dab00
    Asymmetric: yes
    Current Load Balance: round-robin
    Logical Unit Group ID: NA
    Aauto Failback: on
    Auto Probing: NA

Paths:
    Initiator Port Name: 2000000173818713
    Target Port Name: 20030003ba27d212
    Override Path: NA
    Path State: OK
    Disabled: no

    Initiator Port Name: 2000000173018713
    Target Port Name: 20030003ba27d095
    Override Path: NA
    Path State: OK
    Disabled: no

Target Port Group:
    ID: 2
    Explicit Failover: yes
    Access State: standby
    Target Ports:
            Name: 20030003ba27d212
```

例 5 Displaying the Properties of a Particular Logical Unit (続き)

```
            Relative ID: 0

    ID: 5
    Explicit Failover: yes
    Access State: active
    Target Ports
            Name: 20030003ba27d095
            Relative ID: 0
```

例 6 Enabling a Path

The following command uses the `enable path` subcommand to enable the path with the specified initiator port, target port, and logical unit.

```
# mpathadm enable path -i 2000000173018713 -t 20030003ba27d095     \
-l /dev/rdsk/c4t60003BA27D2120004204AC2B000DAB00d0s2
```

例 7 Modifying mpath-support To Turn On autofailback

```
# mpathadm modify mpath-support -a on libmpscsi_vhci.so
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/storage/multipath-utilities
	system/library/storage/libmpapi (`exec_attr(4)` entry)
Interface Stability	Committed

関連項目

`stmsboot(1M)`, `libMPAPI(3LIB)`, `exec_attr(4)`, `attributes(5)`

名前

mpstat – report per-processor or per-processor-set statistics

形式

```
/usr/bin/mpstat [-aq] [-p | -P set] [-T u | d] [interval [count]]
```

機能説明

The `mpstat` command reports processor statistics in tabular form. Each row of the table represents the activity of one processor. The first table summarizes all activity since boot. Each subsequent table summarizes activity for the preceding interval. All values are rates listed as events per second unless otherwise noted.

During execution of the kernel status command, the `state` of the kernel can change. If relevant, a state change message is included in the `mpstat` output, in one of the following forms:

```
<<processor 3 moved from pset: -1 to: 1>>
<<pset destroyed: 1>>
<<pset created: 1>>
<<processors added: 1, 3>>
<<processors removed: 1, 3>>
```

The `mpstat` command reports the following information:

`CPU` or `SET`	Without the `-a` option, `mpstat` reports `CPU` statistics for a processor ID. With the `-a` option, `mpstat` reports `SET` statistics for a processor set ID.
`minf`	minor faults
`mjf`	major faults
`xcal`	inter-processor cross-calls
`intr`	interrupts
`ithr`	interrupts as threads (not counting clock interrupt)
`csw`	context switches
`icsw`	involuntary context switches
`migr`	thread migrations (to another processor)
`smtx`	spins on mutexes (lock not acquired on first try)
`srw`	spins on readers/writer locks (lock not acquired on first try)
`syscl`	system calls
`usr`	percent user time
`sys`	percent system time
`wt`	the I/O wait time is no longer calculated as a percentage of CPU time, and this statistic will always return zero.
`idl`	percent idle time

`sze`	number of processors in the requested processor set
`set`	processor set membership of each CPU

オプション

The following options are supported:

`-a`	Aggregate output by processor set. Sort the output by set. The default output is sorted by CPU number.
`-p`	Report processor set membership of each CPU. Sort the output by set. The default output is sorted by CPU number.
`-P` *set*	Display only those processors in the specified *set*.
`-q`	Suppress messages related to state changes.
`-T u | d`	Specify `u` for a printed representation of the internal representation of time. See `time(2)`. Specify `d` for standard date format. See `date(1)`.
interval	Report once each *interval* seconds.
count	Only print *count* reports.

使用例

例 1　Using mpstat to Generate User and System Operation Statistics

The following command generates processor statistics over a five–second interval in two reports. The command shows the processor set membership of each CPU. The default output is sorted by CPU number, aggregated by *processor set*, for user (`usr`) and system (`sys`) operations.

```
example% mpstat -ap 5 2


SET minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl sze
  0    6   0  355  291  190  22    0    0    0   0    43   0   2  0  43   1
  1   24  17  534  207  200  70    1    0    2   0   600   4   1  0  84   2
  2   19   7  353  325  318  44    0    0    5   0   345   1   1  0  94   3
  3   36   2  149  237  236  14    0    0    4   0    97   0   0  0  98   2
SET minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl sze
  0    1   0  720  405  304  55    0    0   18   0    12   0  15  0  81   1
  1    0  69 1955  230  200 313   33    4   41   9  7086  34  10  0  19   2
  2    0  46  685  314  300 203   11    0   54   1  5287  36   6  0  28   3
  3    0   0   14  386  384   0    0    0    0   0     0   0   0  0 100   2
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	See below.

Invocation is evolving. Human readable output is unstable.

関連項目

sar(1), date(1), iostat(1M), sar(1M), vmstat(1M), time(2), attributes(5)

注意事項

The sum of CPU utilization might vary slightly from 100 due to rounding errors in the production of a percentage figure.

The total time used for CPU processing is the sum of usr and sys output values, reported for user and system operations. The idl value reports the time that the CPU is idle for any reason other than pending disk I/O operations.

Run the iostat command with the -x option to report I/O service times in svc_t output. The iostat utility also reports the same wt, user (us), and system (sy) statistics. See iostat(1M) for more information.

When executing in a zone and if the pools facility is active, mpstat(1M) will only provide information for those processors which are a member of the processor set of the pool to which the zone is bound.

名前

msgid – generate message IDs

形式

/usr/sbin/msgid

機能説明

The msgid utility generates message IDs.

A message ID is a numeric identifier that uniquely identifies a message. Although the probability of two distinct messages having the same ID is high, this can be greatly reduced with the appropriate priority or facility.level designator (see syslogd(1M)). Specifically, the message ID is a hash signature on the message's unexpanded format string, generated by STRLOG_MAKE_MSGID() as defined in <sys/strlog.h>.

syslogd(1M) is a simple filter that takes strings as input and produces those same strings, preceded by their message IDs, as output. Every message logged by syslogd(1M) includes the message ID. The message ID is intended to serve as a small, language-independent identifier.

使用例

例 1 Using the msgid command to generate a message ID

The following example uses the msgid command to generate a message ID for the echo command.

```
example# echo hello | msgid205790 hello
```

例 2 Using the msgid command to generate a message catalog

The following example uses the msgid command to enumerate all of the messages in the binary ufs, to generate a message catalog.

```
example# strings /kernel/fs/ufs | msgid

137713 free:
    freeing free frag, dev:0x%lx, blk:%ld, cg:%d, ino:%lu, fs:%s
567420 ialloccg: block not in mapfs = %s
845546 alloc: %s: file system full
...
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

syslogd(1M), attributes(5), log(7D)

名前

mvdir – move a directory

形式

`/usr/sbin/mvdir` *dirname name*

機能説明

`mvdir` moves directories within a file system. `dirname` must be a directory. If *name* does not exist, it will be created as a directory. If *name* does exist, and is a directory, `dirname` will be created as *name/dirname*. `dirname` and *name* may not be on the same path; that is, one may not be subordinate to the other. For example:

```
example% mvdir x/y x/z
```

is legal, but

```
example% mvdir x/y x/y/z
```

is not.

オペランド

`dirname` The name of the directory that is to be moved to another directory in the filesystem.

name The name of the directory into which `dirname` is to be moved. If *name* does not exist, it will be created. It may not be on the same path as `dirname`.

使用法

See `largefile(5)` for the description of the behavior of `mvdir` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

終了ステータス

`0` Successful operation.

`>0` Operation failed.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`mkdir(1)`, `mv(1)`, `attributes(5)`, `largefile(5)`

名前

named, in.named – Internet domain name server

形式

```
named [-fgsVv] [-c config-file] [-d debug-level] [-m flag]
      [-n #cpus] [-p port] [-S #max-socks] [-t directory]
      [-u user] [-x cache-file] [-4 | -6]
```

機能説明

The `named` utility is a Domain Name System (DNS) server, part of the BIND 9 distribution from ISC. For more information on the DNS, see RFCs 1033, 1034, and 1035.

When invoked without arguments, `named` reads the default configuration file `/etc/named.conf`, reads any initial data, and listens for queries.

`in.named` is a link to `named`.

オプション

The following options are supported:

`-4`
Use only IPv4 transport. By default, both IPv4 and IPv6 transports can be used. Options `-4` and `-6` are mutually exclusive.

`-6`
Use only IPv6 transport. By default, both IPv4 and IPv6 transports can be used. Options `-4` and `-6` are mutually exclusive.

`-c` *config-file*
Use *config-file* as the configuration file instead of the default `/etc/named.conf`. To ensure that reloading the configuration file continues to work after the server has changed its working directory due to to a possible *directory* option in the configuration file, *config-file* should be an absolute pathname.

`-d` *debug-level*
Set the daemon's debug level to *debug-level*. Debugging traces from `named` become more verbose as the debug level increases.

`-f`
Run the server in the foreground (that is, do not run as a daemon).

`-g`
Run the server in the foreground and force all logging to `stderr`.

`-m` *flag*
Turn on memory usage debugging flags. Possible flags are `usage`, `trace`, and `record`, `size`, and `mctx`. These correspond to the `ISC_MEM_DEBUG`*XXXX* flags described in `<isc/mem.h>`.

`-n` *#cpus*
Create *#cpus* worker threads to take advantage of multiple CPUs. If not specified, `named` will try to determine the number of CPUs present and create one thread per CPU. If it is unable to determine the number of CPUs, a single worker thread will be created.

`-p` *port*
Listen for queries on port *port*. If not specified, the default is port 53.

-S *#max-socks*
Allow named to use up to *#max-socks* sockets.

This option should be unnecessary for the vast majority of users. The use of this option could even be harmful, because the specified value might exceed the limitation of the underlying system API. It therefore should be set only when the default configuration causes exhaustion of file descriptors and the operational environment is known to support the specified number of sockets. Note also that the actual maximum number is normally a little smaller than the specified value because named reserves some file descriptors for its internal use.

-s
Write memory usage statistics to *stdout* on exit.

This option is mainly of interest to BIND 9 developers and might be removed or changed in a future release.

-t *directory*
Change the root directory using chroot(2) to *directory* after processing the command line arguments, but before reading the configuration file.

This option should be used in conjunction with the -u option, as chrooting a process running as root does not enhance security on most systems; the way chroot() is defined allows a process with root privileges to escape a chroot jail.

-u *user*
Set the real user ID using setuid(2) to *user* after completing privileged operations, such as creating sockets that listen on privileged ports.

-V
Report the version number and build options, and exit.

-v
Report the version number and exit.

-x *cache-file*
Load data from *cache-file* into the cache of the default view.

Do not use this option. It is of interest only to BIND 9 developers and might be removed or changed in a future release.

拡張機能説明

This section describes additional attributes of named.

SMF Properties

When starting named from the service management facility, smf(5), named configuration is read from the service configuration repository. Use svcprop(1) to list the properties and svccfg(1M) to make changes.

The following application configuration properties are available to administrators:

options/*server*
: Specifies the server executable to be used instead of the default server, /usr/sbin/named.

options/*configuration_file*
: Specifies the configuration file to be used instead of the default, /etc/named.conf. A directory option might be specified in the configuration file. To ensure that reloading the configuration file continues to work in such a situation, *configuration_file* should be specified as an absolute pathname. This pathname should not include the *chroot_dir* pathname. This property is the equivalent of the -c option.

options/*ip_interfaces*
: Specifies over which IP transport, IPv4 or IPv6, BIND will transmit. Possible values are IPv4 or IPv6. Any other setting assumes all, the default. This property is the equivalent of command line option -4 or -6

options/*listen_on_port*
: Specifies the default UDP and TCP port to be used for listening to DNS requests. This property is the equivalent of the command line option -p *port*.

options/*debug_level*
: Specifies the default debug level. The default is 0, which means no debugging. The higher the number the more verbose debug information becomes. Equivalent of the command line option -d *debug_level*.

options/*threads*
: Specifies the number of CPU worker threads to create. The default of 0 causes named to try and determine the number of CPUs present and create one thread per CPU. Equivalent of command line option -n *#cpus*.

options/*chroot_dir*
: Specifies the directory to be used as the root directory after processing SMF properties and the command line arguments but before reading the configuration file. Use this property when using a chroot(2) environment. Synonymous to command line option -t *pathname*.

  When using chroot(2), named is unable to disable itself when receiving rndc(1M) stop or halt commands. Instead, you must use the svcadm(1M) disable command.

In the event of a configuration error originating in one of the above SMF application options, named displays a message providing information about the error and the parameters that need correcting. The process then exits with exit code SMF_EXIT_ERR_CONFIG.

At startup, in the event of an error other than a configuration error, named exits with exit code SMF_EXIT_ERR_FATAL. Both of this code and SMF_EXIT_ERR_CONFIG cause the start method, smf_method(5), to place the service in the maintenance state, which can be observed with the svcs(1) command svcs -x.

In addition to the properties listed above, the following property can be used to invoke named as a user other than root:

start/*user*
: Specifies the identity of the user that is invoking named. See smf_method(5) and chroot(2). Note that the user must have solaris.smf.manage.bind authorization. Without this role the named will be unable to manage its SMF FMRI and named will automatically be restarted by the SMF after an rndc(1M) stop or halt command. See EXAMPLES for a sequence of commands that establishes the correct authorization.

SIGNALS

In routine operation, signals should not be used to control the nameserver; rndc(1M) should be used instead.

SIGHUP
: Force a reload of the server.

SIGINT, SIGTERM
: Shut down the server.

The result of sending any other signals to the server is undefined.

Configuration

The named configuration file is too complex to describe in detail here. A list of configuration options is provided in the named.conf man page shipped with the BIND 9 distribution. A complete description is provided in the *BIND 9 Administrator Reference Manual*.

使用例

例 1 Configuring named to Transmit Only over IPv4 Networks

The following command sequence configures named such that it will transmit only over IPv4 networks.

```
# svccfg -s svc:network/dns/server:default setprop \
> options/ip_interfaces=IPv4
# svcadm refresh svc:network/dns/server:default
#
```

例 2 Listing Current Configuration File and Setting an Alternative File

The following sequence of commands lists the current named configuration file and sets an alternative file.

```
# svcprop -p options/configuration_file dns/server:default
/etc/named.conf
# svccfg -s dns/server:default setprop \
> options/configuration_file=/var/named/named.conf
# svcadm refresh dns/server:default
# svcprop -p options/configuration_file dns/server:default
/var/named/named.conf
```

例 3 Establishing Appropriate Authorization for named

To have named start with the solaris.smf.manage.bind authorization, perform the steps shown below.

Add the user dnsadmin to the solaris.smf.manage.bind role:

例 3 Establishing Appropriate Authorization for named (続き)

```
# usermod -A solaris.smf.manage.bind dnsadmin
Observe effect of command:
# tail -1 /etc/user_attr
dnsadmin::::type=normal;auths=solaris.smf.manage.bind
```

Modify the service properties:

```
# svccfg
svc:> select svc:/network/dns/server:default
svc:/network/dns/server:default> setprop start/user = dnsadmin
svc:/network/dns/server:default> setprop start/group = dnsadmin
svc:/network/dns/server:default> exit
# svcadm refresh svc:/network/dns/server:default
# svcadm restart svc:/network/dns/server:default
```

Because only root has write access to create the default process-ID file, `/var/run/named/named.pid`, `named` must be configured to use an alternative path for the user `dnsadmin`. Here is an example of how to accomplish this:

```
# mkdir /var/named/tmp
# chown dnsadmin /var/named/tmp
```

Shown below is what you must add to `named.conf` to make use of the directory created above.

```
# head /etc/named.conf
options {
directory "/var/named";
pid-file "/var/named/tmp/named.pid";
};
```

ファイル

`/etc/named.conf`
: default configuration file

`/var/run/named/named.pid`
: default process-ID file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dns/bind
Interface Stability	Volatile

関連項目

`svcs(1)`, `named-checkconf(1M)`, `named-checkzone(1M)`, `rndc(1M)`, `rndc-confgen(1M)`, `svcadm(1M)`, `svccfg(1M)`, `svcprop(1)`, `chroot(2)`, `setuid(2)`, `bind(3SOCKET)`, `attributes(5)`, `smf(5)`, `smf_method(5)`

RFC 1033, RFC 1034, RFC 1035

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at `https://www.isc.org/software/bind/documentation`.

The `named.conf` man page shipped with the BIND 9 distribution

名前

named-checkconf – named configuration file syntax checking tool

形式

`named-checkconf [-hjvz] [-t` *directory*`]` *filename*

機能説明

The `named-checkconf` utility checks the syntax, but not the semantics, of a specified configuration file.

オプション

The following options are supported:

`-h`
Display the usage summary and exit.

`-j`
When loading a zonefile, read the journal if it exists.

`-t` *directory*
Change the root directory to *directory* so that include directives in the configuration file are processed as if run by a named configuration whose root directory has been similarly changed.

`-v`
Print the version of the `named-checkconf` program and exit.

`-z`
Perform a test load of the master zones found in `named.conf`.

オペランド

The following operands are supported:

filename
The name of the configuration file to be checked. If not specified, it defaults to `/etc/named.conf`.

終了ステータス

`0`
No errors were detected.

`1`
An error was detected.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dns/bind
Interface Stability	Volatile

関連項目

`named(1M)`, `named-checkzone(1M)`, `attributes(5)`

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at `https://www.isc.org/software/bind/documentation`.

名前

named-checkzone, named-compilezone – zone file validity checking or converting tool

形式

```
named-checkzone [-Ddhjqv] [-c class] [-F format] [-f format]
     [-i mode] [-k mode] [-M mode] [-m mode] [-n mode]
     [-o filename] [-S mode] [-s style] [-t directory]
     [-W mode] [-w directory] zonename filename

named-compilezone [-Ddjqv] [-C mode] [-c class] [-F format]
     [-f format] [-i mode] [-k mode] [-m mode] [-n mode]
     [-o filename] [-s style] [-t directory]
     [-W mode] [-w directory] zonename filename
```

機能説明

The `named-checkzone` utility checks the syntax and integrity of a zone file. It performs the same checks as `named(1M)` does when loading a zone. The `named-checkzone` utility is useful for checking zone files before configuring them into a name server.

`named-compilezone` is similar to `named-checkzone`, differing in that it always dumps the zone contents to a specified file in a specified format. Additionally, it applies stricter check levels by default, since the dump output will be used as an actual zone file loaded by `named(1M)`. Unless manually specified otherwise, the check levels must be at least as strict as those specified in the `named` configuration file.

オプション

For either or both utilities, the following options are supported:

`-c` *class*
: Specify the class of the zone. If not specified, “IN” is assumed.

`-D`
: Dump zone file in canonical format.

`-d`
: Enable debugging.

`-F` *format*
: Specify the format of the output file specified. Possible formats are `text` (default) and `raw`. For `named-checkzone`, this does not cause any effects unless it dumps the zone contents.

`-f` *format*
: Specify the format of the zone file. Possible formats are `text` (default) and `raw`.

`-h`
: Display usage message for `named-checkzone`.

`-i` *mode*
: Perform post-load zone integrity checks. Possible modes are `full` (default), `full-sibling`, `local`, `local-sibling`, and `none`.

  Mode `full` checks that MX records refer to the A or AAAA record (both in-zone and out-of-zone hostnames). Mode `local` checks only MX records that refer to in-zone hostnames.

Mode `full` checks that SRV records refer to the A or AAAA record (both in-zone and out-of-zone hostnames). Mode `local` checks only SRV records that refer to in-zone hostnames.

Mode `full` checks that delegation NS records refer to A or AAAA record (both in-zone and out-of-zone hostnames). It also checks that glue address records in the zone match those advertised by the child. Mode `local` checks only NS records that refer to in-zone hostnames or check that some required glue exists, that is, when the nameserver is in a child zone.

Mode `full-sibling` and `local-sibling` disable sibling glue checks, but are otherwise the same as `full` and `local`, respectively.

Mode `none` disables the checks.

`-k` *mode*
Perform "check-name" checks with the specified failure mode. Possible modes are `fail` (default for `named-compilezone`), `warn` (default for `named-checkzone`) and `ignore`.

`-j`
Read the journal, if it exists, when loading the zone file.

`-M` *mode*
Check if an MX record refers to a `CNAME`. Possible modes are `fail`, `warn` (default) and `ignore`.

`-m` *mode*
Specify whether MX records should be checked to see if they are addresses. Possible modes are `fail`, `warn` (default) and `ignore`.

`-n` *mode*
Specify whether NS records should be checked to see if they are addresses. Possible modes are `fail` (default for `named-compilezone`), `warn` (default for `named-checkzone`) and `ignore`.

`-o` *filename*
Write zone output to *filename*. If filename is `-` (a hyphen), then write to standard out. The hyphen mandatory for `named-compilezone`

`-q`
Run in quiet mode, reporting only the exit status.

`-S` *mode*
Check if a SRV record refers to a `CNAME`. Possible modes are `fail`, `warn` (default) and `ignore`.

`-s` *style*
Specify the style of the dumped zone file. Possible styles are `full` (default) and `relative`. The `full` format is most suitable for processing automatically by a separate script. The `relative` format is more human-readable and is thus suitable for editing by hand. For

named-checkzone this option does not cause any effects unless it dumps the zone contents. It also has no effect if the output format is not text.

-t *directory*
chroot to directory so that include directives in the configuration file are processed as if run by a similarly chrooted named.

-v
Print the version of the named-checkzone program and exit.

-W *mode*
Specify whether to check for non-terminal wildcards. Non-terminal wildcards are almost always the result of a failure to understand the wildcard matching algorithm (RFC 1034). Possible modes are warn (default) and ignore.

-w *directory*
chdir to *directory* so that relative filenames in master file $INCLUDE directives work. This is similar to the directory clause in named.conf.

オペランド

The following operands are supported:

filename
The name of the zone file.

zonename
The domain name of the zone being checked.

終了ステータス

0
No errors were detected.

1
An error was detected.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dns/bind
Interface Stability	Volatile

関連項目

named(1M), named-checkconf(1M), attributes(5)

RFC 1035

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at https://www.isc.org/software/bind/documentation.

名前

ncaconfd – Solaris Network Cache and Accelerator (NCA) configuration daemon

形式

`/usr/lib/inet/ncaconfd [-al ]` *interface1* `[`*interface2* ...`]`

機能説明

Use the `ncaconfd` utility to set up NCA on a system. At boot time, the `ncakmod` initialization script reads in `nca.if(4)` to determine on which interface(s) NCA should run. `ncaconfd` then sets up the interface.

`ncaconfd` also operates as a daemon if the `nca_active` key is set to enabled in `ncakmod.conf(4)` file. In this case, `ncaconfd` will continue as a daemon after all the NCA interfaces have been set up, listening for routing changes. The changes are then passed to NCA to control which interface NCA should use to make active outgoing TCP connections.

オプション

The following options are supported:

`-a` Enable active connections.

`-l` Enable logging.

ファイル

`/etc/nca/ncakmod.conf`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/http-cache-accelerator
Interface Stability	Committed

関連項目

`ncakmod(1)`, `nca.if(4)`, `ncakmod.conf(4)`, `attributes(5)`, `nca(7d)`

名前

ncheck – generate a list of path names versus i-numbers

形式

```
ncheck [-F FSType] [-V] [generic_options]
      [-o FSType-specific_options] [special]...
```

機能説明

`ncheck` with no options generates a path-name versus i-number list of all files on *special*. If *special* is not specified on the command line the list is generated for all *specials* in `/etc/vfstab` which have a numeric `fsckpass`. *special* is the raw device on which the file system exists.

オプション

`-F` Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by finding an entry in the table that has a numeric `fsckpass` field and an `fsckdev` that matches *special*.

`-V` Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from `/etc/vfstab`. This option may be used to verify and validate the command line.

generic_options Options that are commonly supported by most FSType-specific command modules. The following options are available:

- `-i` *i-list* Limit the report to the files on the *i-list* that follows. The *i-list* must be separated by commas with no intervening spaces.
- `-a` Print the names "." and ". ." which are ordinarily suppressed.
- `-s` Report only special files and files with set-user-ID mode. This option may be used to detect violations of security policy.

`-o` Specify *FSType-specific_options* in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the FSType-specific module of the command.

使用法

See `largefile(5)` for the description of the behavior of `ncheck` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/vfstab` list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`vfstab(4)`, `attributes(5)`, `largefile(5)` Manual pages for the FSType-specific modules of `ncheck`

注意事項 This command may not be supported for all *FSTypes*.

名前

ncheck_ufs – generate pathnames versus i-numbers for ufs file systems

形式

`ncheck -F ufs [generic_options] [-o m] [special]...`

機能説明

`ncheck -F ufs` generates a pathname versus i-number list of files for the `ufs` file system residing on *special*. Names of directory files are followed by '`/.`'.

オプション

See `ncheck(1M)` for the list of *generic_options* supported.

`-o` Specify `ufs` file system specific options. The available option is:

`m` Print mode information.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`ff(1M)`, `ncheck(1M)`, `attributes(5)`

診断

When the file system structure is improper, '`??`' denotes the "parent" of a parentless file and a pathname beginning with '`. . .`' denotes a loop.

名前

ndd – get and set driver configuration parameters

形式

`ndd [-set] driver parameter [value]`

機能説明

ndd gets and sets selected configuration parameters in some kernel drivers. Currently, ndd only supports the drivers that implement the TCP/IP Internet protocol family. Each driver chooses which parameters to make visible using ndd. Since these parameters are usually tightly coupled to the implementation, they are likely to change from release to release. Some parameters may be read-only.

注 – It is strongly encouraged that you use ipadm(1M), rather than ndd, to modify or retrieve TCP/IP Internet protocols. The current ndd command will be made obsolete in a future release, replaced by ipadm(1M). Please see NOTES for more information.

If the ndd -set option is omitted, ndd queries the named *driver*, retrieves the value associated with the specified *parameter*, and prints it. If the -set option is given, ndd passes *value*, which must be specified, down to the named *driver* which assigns it to the named *parameter*.

By convention, drivers that support ndd also support a special read-only *parameter* named "?" which can be used to list the parameters supported by the driver.

使用例

例 1 Getting Parameters Supported By The TCP Driver

To see which parameters are supported by the TCP driver, use the following command:

```
example% ndd /dev/tcp \?
```

The parameter name "?" may need to be escaped with a backslash to prevent its being interpreted as a shell meta character.

The following command sets the value of the parameter *ip_forwarding* in the dual stack IP driver to zero. This disables IPv4 packet forwarding.

```
example% ndd -set /dev/ip ip_forwarding 0
```

Similarly, in order to disable IPv6 packet forwarding, the value of parameter *ip6_forwarding*

```
example% ndd -set /dev/ip ip6_forwarding 0
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

dladm(1M), ipadm(1M), ioctl(2), attributes(5), nca(7d), arp(7P), ip(7P), ip6(7P), tcp(7P), udp(7P)

注意事項

The parameters supported by each driver may change from release to release. Like programs that read `/dev/kmem`, user programs or shell scripts that execute `ndd` should be prepared for parameter names to change.

The `ioctl()` command that `ndd` uses to communicate with drivers is likely to change in a future release. User programs should avoid making dependencies on it.

The use of `ndd` to administer Layer 2 (Data Link layer) drivers is strongly discouraged as this capability is to be obsoleted in a future release, replaced by `dladm(1M)`. Please refer to the driver-specific man page in section 7D of the SunOS man pages.

The use of `ndd` to administer the drivers that implement the TCP/IP Internet protocol family (IP/TCP/SCTP/UDP/ICMP) is strongly discouraged as this capability is to be obsoleted in a future release, replaced by `ipadm`. Please see `ipadm(1M)` for instructions for modifying and retrieving supported protocol properties.

The meanings of many `ndd` parameters make sense only if you understand how the driver is implemented.

名前

ndmpadm – NDMP (Network Data Management Protocol) の動作の管理

形式

`/usr/sbin/ndmpadm [-? ] subcommand [options] [direct-object]`

機能説明

`ndmpadm` コマンドを使用して `ndmpd(1M)` デーモンへ照会を行い、アクティブなセッションのステータスの取得、セッションの終了、バックアップデバイスの照会、および現在の NDMP (Network Data Management Protocol) サービスの変数やプロパティーの設定と取得を行うことができます。`ndmpadm` は、サブコマンドのセットとして実装されており、その多くは独自の直接オブジェクトを保持します。詳細については、各サブコマンドのセクションを参照してください。一部のサブコマンドでは、オプションをサポートしています。詳細については、該当するサブコマンドの説明を参照してください。

`ndmpadm` コマンドは、次のサブコマンドをサポートします。

`disable`
: 指定された認証パスワード処理を無効にします。

`enable`
: 指定された認証パスワード処理を有効にします。

`get`
: NDMP 構成プロパティーの値を取得します。

`kill-sessions`
: アクティブなセッションを終了します。

`set`
: NDMP 構成プロパティーの値を設定します。

`show-devices`
: サーバーに接続されているテープデバイスの一覧を取得します。

`show-sessions`
: アクティブな NDMP セッションの詳細を表示します。

オプション

次のオプションがサポートされています。

`-?` すべてのサブコマンドとオプションの一覧を表示します。

サブコマンド

`ndmpadm` コマンドは、次のサブコマンドをサポートします。

disable サブコマンド

`disable` サブコマンドの構文は次のとおりです。

```
# ndmpadm disable -a auth-type
```

このサブコマンドは、NDMP クライアントのリモートアクセスに関して *auth-type* で指定された認証タイプを無効にします。*auth-type* の有効な値は、`cram-md5` または `cleartext` です。

enable サブコマンド

enable サブコマンドの構文は次のとおりです。

```
# ndmpadm enable -a auth-type -u username
```

このサブコマンドは、確認のためにユーザーにパスワードの入力を 2 回求めます。その後、NDMP クライアントアクセス用に指定されたユーザー名とパスワードを使って、指定された認証タイプをアクティブにします。*auth-type* の有効な値は、cram-md5 または cleartext です。

get サブコマンド

get サブコマンドの構文は次のとおりです。

```
# ndmpadm get [-p] [property] [[-p] property=value]...
```

プロパティー名は、set サブコマンドで使用されるものと同じです。これらについては後述します。プロパティーを指定しない場合は、get サブコマンドによりすべての構成プロパティーが返されます。

kill-sessions サブコマンド

kill-sessions サブコマンドを使って、*ID* に指定された番号のセッションを終了できます。

kill-sessions サブコマンドの構文は次のとおりです。

```
# ndmpadm kill-sessions ID
```

set サブコマンド

set サブコマンドの構文は次のとおりです。

```
# ndmpadm set [-p] property=value [[-p] property=value]...
```

set サブコマンドを使って設定可能なプロパティーについては、ndmp(4) のマニュアルページを参照してください。

show-devices サブコマンド

show-devices サブコマンドの構文は次のとおりです。

```
# ndmpadm show-devices
```

このサブコマンドは、システムに現在接続されているテープドライブおよびライブラリの名前、製造元、シリアル番号、その他の情報を一覧表示します。

show-sessions サブコマンド

show-sessions サブコマンドは、セッションの詳細を表示します。show-sessions サブコマンドの構文は次のとおりです。

```
# ndmpadm show-sessions [-i tape,scsi,data,mover] [ID]
```

show-sessions サブコマンドでは次の引数をサポートしています。

-i tape,scsi,data,mover
: データを取得するインタフェースのタイプを特定します。インタフェースを指定しない場合、show-sessions ではすべてのインタフェースタイプの情報が表示されます。

ID
データを表示する特定のセッションを指定します。*ID* を指定しない場合は、`show-sessions` ではすべてのセッションのデータが表示されます。

使用例

例 1 すべての NDMP 接続のステータスを取得する

次のコマンドは、すべての接続のステータスを取得します。

```
# ndmpadm show-devices
```

例 2 特定の接続タイプのステータスを取得する

次のコマンドは、テープおよび SCSI インタフェースのステータスを取得します。

```
# ndmpadm show-sessions -i scsi,tape
```

例 3 プロトコルのバージョンを制限する

次のコマンドは、使用する NDMP プロトコルをバージョン 3 に制限します。

```
# ndmpadm set -p version=3
```

例 4 現在のバージョン番号を取得する

次のコマンドは、現在実行中の NDMP のバージョン番号を取得します。

```
# ndmpadm get -p version
```

例 5 特定のセッションを接続解除する

次のコマンドはセッション 5 を接続解除します。セッション番号は、以前に `ndmpadm show-sessions` コマンドから取得した番号です。

```
# ndmpadm kill-session 5
```

例 6 NDMP プロパティーの値をすべて取得する

次のコマンドは、NDMP プロパティーの値をすべて取得します。

```
# ndmpadm get
```

例 7 CRAM-MD5 認証を有効にする

次のコマンドは、CRAM-MD5 認証を有効にします。

```
# ndmpadm enable -a cram-md5 -u admin
Enter new password:*****
Re-enter password:*****
```

例 8 平文パスワード認証を無効にする

次のコマンドは、平文パスワード認証を無効にします。

```
# ndmpadm disable -a cleartext
```

終了ステータス

`0` 正常終了。

1. エラー (ndmpd デーモンが実行されていないなど) が発生したため、ndmpadm がデーモンと通信できませんでした。
2. 無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/ndmp
インタフェースの安定性	確実

関連項目

dump(1), tar(1), ndmpd(1M), ndmpstat(1M), svccfg(1M), syslogd(1M), ndmp(4), attributes(5), smf(5)

注意事項

ndmpd(1M) デーモンは、サービス管理機能 (smf(5)) によって、次のサービス識別子として管理されます。

```
svc:/system/ndmpd
```

プロパティーの設定や取得など、このサービスに関する管理操作は、svccfg(1M) を使用しても実行できます。たとえば、次の方法で DAR (Direct Access Recovery) モードを有効にできます。

```
# svccfg -s svc:/system/ndmpd
svc:/system/ndmpd> setprop ndmpd/dar-support = yes
```

プロパティーの一覧を取得するには、次のようにします。

```
# svccfg -s svc:/system/ndmpd
svc:/system/ndmpd> listprop
```

名前

ndmpd – NDMP (Network Data Management Protocol) デーモン

形式

/usr/lib/ndmp/ndmpd

機能説明

ndmpd デーモンは、クライアント NDMP (Network Data Management Protocol) の要求を処理します。NDMP は、バックアップや復旧に使用される、オープンな企業規模のネットワークベースデータ管理プロトコルです。ndmpd デーモンにより、ユーザーは DMA (Data Management Application) クライアントを使用してデータのバックアップと復旧を管理できるようになります。NDMP プロトコルを使って、データの移動を調整したり、DMA と NDMP サーバー間または 2 台の NDMP サーバー間を制御したりできます。

デフォルトでは、ndmpd は無効になっています。

終了ステータス

0 正常終了。

1 プロセスのフォークや相互排他ロックの初期化に失敗するなどのエラーが発生したために、ndmpd デーモンの初期化が妨げられました。

2 無効なコマンド行オプションが指定されました。

ファイル

/usr/lib/ndmp/ndmpd
ネットワークデータ管理プロトコルサーバーのバイナリ。

/var/ndmp/ndmp.log
ネットワークデータ管理プロトコルのログメッセージファイル。このファイルは、リブート時に削除されます。

/var/ndmp/dumpdates
ダンプバックアップの日付とレベルに関する情報を格納するテキストファイル。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/ndmp
インタフェースの安定性	確実

関連項目

svcs(1), ndmpadm(1M), svcadm(1M), svccfg(1M), syslogd(1M), attributes(5), smf(5)

注意事項

ndmpd デーモンは、サービス管理機能 (smf(5)) によって、次のサービス識別子として管理されます。

```
svc:/system/ndmpd:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

ndmpstat – NDMP バックアップ進行状況統計の表示

形式

```
ndmpstat [tapes] [interval [count]]
```

機能説明

ndmpstat ユーティリティーは、NDMP (Network Data Management Protocol) の統計を報告します。この統計には、NDMP ワーカースレッド、ディスク入出力、テープ入出力、ファイル操作、パフォーマンス、およびバックアップ操作が含まれています。

ndmpstat は、すべてのテープおよびディスクの集約的統計を報告します。特定のテープデバイスの統計を取得するには、テープデバイス名を引数としてユーティリティーに渡します。

ndmpstat を呼び出すと、NDMP サービスが呼び出されてからの NDMP デーモンの動作の要約が 1 行で表示され始めます。

表示フィールド

ndmpstat の出力フィールドは次のとおりです。

wthr
: 次の 4 つの各状態にあるワーカースレッドの数を報告します。

  r
  : 実行中のワーカースレッドの数

  w
  : 入出力やページングなどのリソースを待機中の、ブロックされているワーカースレッドの数

  b
  : 実行中のバックアップ操作の数

  r
  : 実行中の復元操作の数

file
: ファイルシステムの使用状況について報告します。

  rd
  : 読み込まれているファイルの数

  wr
  : 書き込まれているファイルの数

disk
: 間隔あたりのディスク操作の数を報告します。

  rd
  : 読み込まれているディスクブロックの数

  wr
  : 書き込まれているディスクブロックの数

`tape`
間隔あたりのテープ操作の数を報告します。1 つの数字で示された、最大 4 つのテープのスロットがあります。数字は、`/dev/rmt` にあるデバイスの名前を示します。

`rd`
読み込まれているテープブロックの数

`wr`
書き込まれているテープブロックの数

`bytes`
転送済みのバイト数を報告します。これは、テープデバイスとディスクデバイスの合計値です。数値は K バイト単位です。

`rd`
読み込まれた K バイト数

`wr`
書き込まれた K バイト数

`perf`
バックアップ/復元操作のパフォーマンスの概算を、秒あたりの M バイト数で表示します。

`bk`
バックアップのパフォーマンス

`rs`
復元のパフォーマンス

`prcnt`
リソースの使用量の割合 (%) を表示します。

`dsk`
ディスクの入出力時間

`tpe`
テープの入出力時間

`otr`
その他の時間 (メモリーまたはアイドル)

「使用例」を参照してください。

オペランド

次のオペランドがサポートされています。

count 統計表示を繰り返す回数を指定します。

tape 優先して出力するテープを指定します。コマンド行にはテープデバイスを 4 台まで指定できます。一般的なテープ名は `/dev/rmt/`*n* で、*n* は整数です。

interval ndmpstat がアクティビティーを集計する秒数を指定します。指定した間隔は、コマンドが終了するまで有効です。

使用例

例 1 ndmpstat を使用する

次のコマンドは、NDMP デーモンの動作の要約を 5 秒間隔で表示します。

```
example% ndmpstat 5
wthr     file      disk      tape        bytes      perf       prcnt
r w b r rd  wr    rd  wr   rd    wr     rd    wr   bk  rs  dsk  tpe  otr
1 0 3 6 50   9   1250  0  32544 4455   42335 3234   5  4   20   40   40
1 0 0 1  1   0    128  0    0   128     64    64    1  0    0   80   20
1 0 0 1  2   0    128  0    0     0     64     0    1  0   80    0   20
1 0 0 1  1   0    128  0    0     0     64     0    1  0   80    0   20
1 0 0 1  3   0    128  0    0     0     64     0    0  0   80    0   20
1 0 0 1  1   0    128  0    0   128     64    64    1  0    0   80   20
^C
example%
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/ndmp
インタフェースの安定性	下記を参照

この呼び出しは開発中です。人間が読める形式の出力は不安定です。

関連項目

iostat(1M), mpstat(1M), ndmpd(1M), ndmpadm(1M), attributes(5)

注意事項

パフォーマンスの数値は正確ではなく、MB/秒単位で切り上げられます。

名前

netadm – ネットワーク構成プロファイルの管理

形式

```
netadm enable [ -p profile-type ] [ -c ncu-class ] profile-name

netadm disable [ -p profile-type ] [ -c ncu-class ] profile-name

netadm list [ -x ] [ -p profile-type ] [ -c ncu-class ]
     [ profile-name ]

netadm show-events

netadm scan-wifi linkname

netadm select-wifi linkname

netadm help
```

機能説明

`netadm` ユーティリティーは、ネットワークプロファイルを管理し、NWAM デーモンを操作するために使用します。

ネットワークプロファイルには、ネットワーク構成プロファイル (NCP)、場所、および外部ネットワーク修飾子 (ENM) の 3 つのタイプがあります。

システムには常に、1 つのアクティブな NCP と 1 つのアクティブな場所があります。別の NCP または場所を有効にすると (activation-mode が `manual`)、現在アクティブな NCP または場所が暗黙的に無効になります。現在の場所 (その activation-mode が `manual` である場合) を無効にすることもできます。ただし、この効果は、ネームサービスなど、システムのネットワーキング機能の一部を「無効」にすることです。NCP を無効にすると、事実上、システムの基本ネットワーク接続がシャットダウンされるので、明示的な無効化は許可されていません。NCP は、別の NCP が有効になったときにのみ暗黙的に無効になります。

逆に、ENM は、常に 0 個以上をアクティブにすることができます。したがって、1 つの ENM を有効または無効にしても、ほかのアクティブな ENM には影響しません。

NCU は個別に有効にしたり無効にしたりすることもできます。指定された NCU は、現在アクティブになっている NCP の一部である必要があり、その activation-mode は `manual` に設定されている必要があります。NCU クラスが指定されない場合、指定された名前を持つすべての NCU (1 つのリンクまたは 1 つのインタフェース、あるいはその両方) が有効化または無効化されます。

オブジェクトの有効化と無効化は非同期に実行されます。したがって、有効化または無効化の要求は成功しても、操作自体が失敗する場合があります。この種の失敗はオブジェクト状態に反映されます。`maintenance` 状態は、最後に行われた操作が失敗したことを示します。

サブコマンド

次のサブコマンドがサポートされています。

`enable` [`-p` *profile-type*] [`-c` *ncu-class*] *profile-name*

指定されたプロファイルを有効にします。プロファイル名が一意でない場合、有効にするプロファイルを識別するためにプロファイルタイプを指定する必要があります。プロファイルタイプが NCU であり、名前が一意でない (つまり、同じ名前のリンク NCU とインタフェース NCU の両方がある) 場合、`-c` オプションを使用して NCU クラスを指定しないかぎり、両方の NCU が有効化されます。プロファイルタイプは、`ncp`、`ncu`、`loc`、または `enm` のいずれかにする必要があります。NCU クラスは、`phys` または `ip` のどちらかにする必要があります。

`disable` [`-p` *profile-type*] [`-c` *ncu-class*] *profile-name*

指定されたプロファイルを無効にします。プロファイル名が一意でない場合、無効にするプロファイルを識別するためにプロファイルタイプを指定する必要があります。プロファイルタイプが NCU であり、名前が一意でない (つまり、同じ名前のリンク NCU とインタフェース NCU の両方がある) 場合、`-c` オプションを使用して NCU クラスを指定しないかぎり、両方の NCU が無効化されます。プロファイルタイプは、`ncu`、`loc`、または `enm` のいずれかにする必要があります。NCU クラスは、`phys` または `ip` のどちらかにする必要があります。

`list` [`-x`] [`-p` *profile-type*] [`-c` *ncu-class*] [*profile-name*]

すべての使用可能なプロファイルとその現在の状態を一覧表示します。名前によって特定のプロファイルが指定された場合、そのプロファイルの現在の状態のみが一覧表示されます。プロファイル名が一意でない場合、指定された名前を持つすべてのプロファイルが一覧表示されます。プロファイルタイプまたは NCU クラス、あるいはその両方を含めると、特定のプロファイルを識別できます。タイプだけが指定された場合、そのタイプのすべてのプロファイルが一覧表示されます。アクティブな NCP を一覧表示すると、その NCP を構成する NCU が表示されます。

`-x` オプションを使用すると、`list` サブコマンドは、デフォルトの表示に含まれる最初の 3 つの列ヘッダー `TYPE`、`PROFILE`、および `STATE` のあとに、ヘッダー `AUXILIARY STATE` の付いた 4 列目の出力を表示します。`AUXILIARY STATE` 列には、プロファイルが指定された状態にある理由が表示されます。

考えられる `STATE` 値は次のとおりです。

`disabled`

まだアクティブ化されていない、手動でアクティブ化されるプロファイル。

`offline`

まだアクティブ化されていない、条件またはシステムによってアクティブ化されるプロファイル。その条件が満たされていなかったために、アクティブになっていない可能性があります。または、より具体的な条件が満たされている別のプロファイルが存在し、これが代わりにアクティブ化されている可能性もあります (場所など、一度にひとつずつアクティブ化する必要があるプロファイルタイプの場合)。

`online`
条件が満たされアクティブ化に成功した、条件またはシステムによってアクティブ化されるプロファイル。または、ユーザーの要求でアクティブ化に成功した、手動でアクティブ化されるプロファイル。

`maintenance`
プロファイルのアクティブ化が試行されましたが、失敗しました。

`initialized`
このプロファイルは、操作がまだ行われていない有効な構成オブジェクトを表します。

`uninitialized`
このプロファイルは、削除された物理リンクに対応する NCU など、システムに存在しない構成オブジェクトを表します。

`show-events`
NWAM デーモンからイベントのストリームを待機し、それらを表示します。

`scan-wifi` *linkname*
リンク linkname に対して無線スキャンを開始します。

`select-wifi` *linkname*
リンク linkname に対するスキャン結果から、接続する無線ネットワークを選択します。必要に応じて、選択や WiFi キーなどの入力が求められます。

`help`
各サブコマンドの簡単な説明を含む使用法に関するメッセージを表示します。

使用例

例 1 ユーザー指定の場所の有効化

次のコマンドは、ユーザー指定の場所を有効にします。

```
# netadm enable -p loc office
Disabled loc 'home'.
Enabled loc 'office'
```

例 2 ENM の無効化

次のコマンドは ENM を無効にします。

```
# netadm disable -p enm myvpn
Disabled enm 'myvpn'.
```

例 3 すべての NCP の一覧表示

次のコマンドはすべての NCP を一覧表示します。

```
# netadm list -xp ncp
TYPE         PROFILE        STATE          AUXILIARY STATE
ncp          Automatic      disabled       disabled by administrator
ncp          User           online         active
```

例 3　すべての NCP の一覧表示　　*(続き)*

```
ncu:phys     nge0          online        interface/link is up
ncu:ip       nge0          online        interface/link is up
ncu:phys     nge1          offline       interface/link is down
ncu:ip       nge1          offline       conditions for activation are
                                         unmet
```

例 4　アクティブな NCP の NCU の一覧表示

次のコマンドは、アクティブな NCP 内のすべての `ip` NCU を一覧表示します。

```
# netadm list -c ip
TYPE        PROFILE        STATE
ncu:ip      bge0           online
ncu:ip      bge1           disabled
```

例 5　スキャンの強制

次のコマンドは、無線インタフェース `wpi0` に対してスキャンを強制します。

```
# netadm scan-wifi wpi0
```

例 6　WiFi ネットワークの選択

次のコマンドは、ESSID をブロードキャストしている WiFi ネットワークを選択します。

```
# netadm select-wifi wpi0
1: ESSID testing BSSID 0:40:96:29:e9:d8
2: ESSID sunwifi BSSID 0:b:e:9f:b5:80
3: ESSID sunwifi BSSID 0:b:e:85:26:c0
4: ESSID sunwifi BSSID 0:b:e:49:2f:80
5: Other

Choose WLAN to connect to [1-5]: 2
#
```

例 7　WiFi ネットワークの選択 (代替)

次のコマンドは、ESSID をブロードキャストしていない WiFi ネットワークを選択します。

```
# netadm select-wifi wpi0
1: ESSID testing BSSID 0:40:96:29:e9:d8
2: ESSID sunwifi BSSID 0:b:e:85:26:c0
3: ESSID sunwifi BSSID 0:b:e:9f:b5:80
4: ESSID sunwifi BSSID 0:b:e:49:2f:80
5: ESSID sunwifi BSSID 0:b:e:49:62:c0
6: Other
```

例 7　WiFi ネットワークの選択 (代替)　　(続き)

```
Choose WLAN to connect to [1-6]: 6

Enter WLAN name: oraclewifi
1: None
2: WEP
3: WPA
Enter security mode: 2

Enter WLAN key for ESSID oraclewifi: 123456

Enter key slot [1-4]: 1
#
```

例 8　nwamd の監視

次のコマンドは、場所を切り換えるときに nwamd(1M) を監視します。

```
# netadm show-events
EVENT           DESCRIPTION
OBJECT_ACTION   loc Automatic -> action refresh
OBJECT_STATE    loc Automatic -> state offline*, method/service executi
OBJECT_STATE    loc Automatic -> state online, active
OBJECT_ACTION   loc home -> action refresh
OBJECT_ACTION   loc NoNet -> action refresh
OBJECT_ACTION   loc User -> action refresh
OBJECT_ACTION   loc home -> action enable
OBJECT_STATE    loc home -> state offline*, method/service executing
OBJECT_STATE    loc Automatic -> state offline, conditions for activati
OBJECT_STATE    loc home -> state online, active
^C
#
```

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

netcfgd(1M), netcfg(1M), nwamd(1M), attributes(5)

JDS/GNOME マニュアルページコレクションで利用可能な nwam-manager(1M) も参照してください。

名前

netcfg – ネットワーク構成プロファイルの作成と変更

形式

```
netcfg
netcfg subcommand [options...]
netcfg [-d] -f command-file
netcfg help [subcommand]
```

機能説明

netcfg ユーティリティーは、システムネットワーク構成プロファイルを操作します。netcfg は対話形式で個々のサブコマンドを使って呼び出すことも、一連のサブコマンドが含まれているコマンドファイルを指定して呼び出すこともできます。

netcfg ユーティリティーは、いくつかの異なるタイプの構成プロファイルで動作します。

- ネットワーク構成プロファイル (NCP)
- 場所
- 外部ネットワーク修飾子 (ENM)
- 既知の WLAN

これらのプロファイルタイプの詳細は、「プロファイル」の節を参照してください。

netcfg コマンドは、スコープ内で実行されます。グローバル、プロファイル、NCP の 3 つのスコープがあります。netcfg が引数なしで呼び出されると、編集セッションがグローバルスコープ内で開始します。グローバルスコープでは、NCP、場所、および ENM プロファイルと、既知の WLAN エントリを操作対象として利用できます。NCP を選択すると、編集セッションが NCP スコープに移動します。そこで、個々の NCU を作成したり選択したりすると、プロファイルスコープに移動します。また、グローバルスコープで、場所、ENM、または既知の WLAN を選択または作成すると、編集セッションがプロファイルスコープに移ります。

指定したプロファイルスコープ内で、プロファイルのプロパティーを表示して変更できます。

対話型モードでは、確定処理が呼び出されるまで、変更は永続ストレージに保存されません。確定処理は「最後」または「終了」時に暗黙的に呼び出されますが、ユーザーが明示的に呼び出すこともできます。確定処理が呼び出されると、プロファイル全体が確定します。永続ストレージの整合性を維持するために、確定処理には検証手順が含まれています。検証が失敗すると、確定処理も失敗します。暗黙の確定処理が失敗した場合、ユーザーは、現在の変更を確定せずに終了するか、または現在のスコープのままさらに変更を続けるオプションが与えられます。

プロファイル

NWAM サービスは、必要なプロパティー値をプロファイルに保存することにより、ネットワーク構成を管理します。次に、どの時点でどのプロファイルをアクティブにするべきかを現在のネットワーク条件に応じて判定し、そのプロファイルをアクティブ化します。前の節で説明したネットワーク構成プロファイル (NCP) に加え、nwamd は場所と ENM のプロファイルも管理します。

ネットワーク構成プロファイル (NCP)

NCP は、物理的リンク、IP トンネルリンク、IP インタフェースといったローカルネットワークのコンポーネントの構成を指定します。IP インタフェースには、どちらかのタイプの基になるリンクを関連付ける必要があります。これらのコンポーネントをまとめて「ネットワーク構成ユニット」または「NCU」と呼びます。

システムには NCP をいくつでも構成できます。Automatic NCP は nwamd によって作成され管理されるため、ユーザーが変更することはできません。この NCP は、システムに存在する各物理リンクに関する 1 つのリンク NCU と 1 つのインタフェース NCU から成ります。リンクがシステムに追加またはシステムから削除されると、それに対応する NCU が Automatic NCP に追加または Automatic NCP から削除されます。この NCP に実装されているポリシーでは、有線リンクが無線リンクよりも優先され、接続されているすべての有線リンクに、あるいは有線リンクが接続されていない場合は 1 つの無線リンクに IP が plumb されます。

また、古い /etc/nwam/llp ファイルに含まれている構成に基づいて、NWAM サービスを旧バージョンからアップグレードする際に User NCP も作成されます。たいてい、アップグレード時に NWAM サービスがアクティブになっていた場合は、User NCP が作成されても、まず Automatic NCP がアクティブになります。ただし、静的 IP アドレスなどの明示的な変更が古い構成に見つかった場合は、まず User NCP がアクティブになります。

最後に、ユーザーは追加の NCP をいくつでも作成できます。これらの NCP はすべてユーザーが管理します。NCU の追加や削除は明示的に行う必要があります。また、現在システムにインストールされているどのリンクにも対応しない NCU を追加することも、システムに存在するリンクに対応している NCU を削除することもできます。ユーザーはこれらの NCP のポリシーを決定できます。

システムは最初に NCP を選択したあとは、アクティブな NCP を変更することはありません。ユーザーは、GUI または netadm(1M) コマンドを使用して、いつでもこれを行うことができます。

場所

場所は、ネームサービス、ドメイン、IP フィルタ、IPsec 構成といったシステム全体のネットワーク構成を指定します。

外部ネットワーク修飾子 (ENM)

外部ネットワーク修飾子は、その名前が示すとおり、ネットワーク構成の変更や作成を行うことのできる NWAM サービス外部のアプリケーションです。 nwamd は、ENM プロファイルの一部として指定されている条件に応じて、ENM をアクティブ化または非アクティブ化します。または、必要に応じてユーザーが手動で ENM をアクティブ化または非アクティブ化することもできます。

場所プロファイルでは、現在のネットワークの状況に基づいて、特定のネットワーク関連サービスセットを自動的に構成できますが、そのサービスセットは制限されています。ENM にはさらなる柔軟性が備わっているため、ユーザーは SMF サービスのプロパティーまたは状態 (あるいはその両方) やほかのシステム設定に対する変更が特定の条件下で適用されるように指定することができます。

プロパティー

`netcfg` は、次のタイプのプロパティーをサポートしています。

- NCU のプロパティー
- インタフェース NCU のプロパティー
- すべてのリンク NCU に共通のプロパティー
- 場所のプロパティー
- ENM のプロパティー
- 既知の WLAN のプロパティー

これらのプロパティーについては、次の項目で説明します。

NCU のプロパティー

次のプロパティーはすべての NCU に共通しています。

`type`: 列挙値: *link* | *interface*
: NCU タイプを、*link* または *interface* のどちらかに指定します。値は指定したクラスに基づいて暗黙的に決まります。

`class`: 列挙値: *phys* (リンク NCU の場合); *ip* (インタフェース NCU の場合)
: NCU クラスを指定します。

`parent`: 文字列: 親 *NCP* の名前
: この NCU がコンポーネントになっている NCP を指定します。

`type`、`class`、および `parent` プロパティーは、NCU の作成時に設定され、あとから変更することはできません。

`enabled`: ブール型: `true` | `false`
: activation-mode が `manual` の場合、enabled プロパティーは NCU の現在の状態を反映します。このプロパティーは読み取り専用です。`netadm(1M)` を使って NCU を有効または無効にすることで、間接的に変更されます。

  デフォルト値は `true` です。

インタフェース NCU のプロパティー

`ip-version`: 列挙値のリスト: `ipv4` | `ipv6`
: この NCU で使用すべき IP のバージョン。

  デフォルト値は `ipv4,ipv6` です。

`ipv4-addrsrc`: 列挙値のリスト: `dhcp` | `static`
: この NCU に割り当てられた IPv4 アドレスのソースを識別します。複数の値を割り当てることができます。割り当てられた値のいずれかが `static` である場合、`ipv4-addr` プロパティーには、NCU に割り当てられる IPv4 アドレスを少なくとも 1 つ含める必要があります。

デフォルト値は dhcp です。

ipv4-addr: IPv4 アドレスのリスト
この NCU に割り当てられる 1 つまたは複数の IPv4 アドレス。

ipv4-default-route:*IPv4* アドレス
デフォルトルーターの IPv4 アドレス。このプロパティーが設定されている場合、IPv4 トラフィックのデフォルトルートは、起動時にこのインタフェースに関連付けられます。

ipv6-addrsrc: 列挙値のリスト: dhcp | autoconf | static
この NCU に割り当てられた IPv6 アドレスのソースを識別します。複数の値を割り当てることができます。割り当てられたいずれかの値が static である場合、ipv6-addr プロパティーには、NCU に割り当てられる IPv6 アドレスを少なくとも 1 つ含める必要があります。

デフォルト値は dhcp,autoconf です。

ipv6-addr: IPv6 アドレスのリスト
この NCU に割り当てられる 1 つまたは複数の IPv6 アドレス。

ipv6-default-route: *IPv6* アドレス
デフォルトルーターの IPv6 アドレス。このプロパティーが設定されている場合、IPv6 トラフィックのデフォルトルートは、起動時にこのインタフェースに関連付けられます。

すべてのリンク NCU に共通のプロパティー

activation-mode: 列挙値: manual | prioritized
この NCU の自動有効化のトリガーのタイプ。

デフォルト値は manual です。

priority-group: uint64: *group priority number*
activation-mode が prioritized に設定されている場合、このプロパティーは、この NCU が所属する優先順位グループを指定します。グループは、1 つまたは複数の NCU から構成されます。数値が小さいほど優先順位は高くなります。使用可能と判断された優先順位がもっとも高いグループは、nwamd(1M) によってアクティブ化され、使用可能でなくなるまで、または優先順位がさらに高いグループが使用可能になるまで、使用され続けます。

priority-mode: 列挙値: exclusive | shared | all
activation-mode が prioritized に設定されている場合、このプロパティーは、優先順位グループの可用性とアクティブ化動作を判断するために使用されるモードを指定します。

exclusive
グループを使用可能にするには、少なくとも 1 つの NCU が使用できる必要があります。1 つの NCU だけがアクティブ化されます。複数のメンバー NCU が使用できる場合は、nwamd(1M) がアクティブ化する NCU をランダムに選択します。

`shared`
グループを使用可能にするには、少なくとも 1 つの NCU が使用できる必要があります。すべての使用可能な NCU がアクティブ化されます。

`all`
グループを使用可能にするには、すべてのグループメンバー NCU が使用できる必要があります。すべてのメンバー NCU がアクティブ化されます。

`link-mac-addr`: 文字列: 48 ビット MAC アドレス
このリンクに割り当てられた MAC アドレス。デフォルトで、NWAM は、リンクに対する工場割り当てのまたはデフォルトの MAC アドレスを要求します。ここで別の値を設定すると、選択されている値が上書きされます。

`link-autopush`: 文字列のリスト: リンクでプッシュされるモジュール
リンクが開かれたときに、リンクを介して自動的にプッシュするモジュールのリストを識別します。

`link-mtu`: `uint64`: このリンクの MTU サイズ
このプロパティーは、自動的に、物理リンクのデフォルトの MTU に設定されます。この値は、このプロパティーを別の値に設定することによって上書きできます。

許可される MTU 値の範囲は、基礎となるハードウェアによって異なります。ドライバをインストールして、基礎となるハードウェアを準備する前に NCU を作成できるので、NCU が編集される時点では設定値を確認することはできません。無効な MTU サイズのために NCU をアクティブ化できない場合は、システムログにこのことが記され、NCU は保守状態に置かれます。

場所のプロパティー

`activation-mode`: 列挙値: `manual` | `system` | `conditional-all` | `conditional-any`
この場所の自動アクティブ化のトリガーのタイプ。

デフォルト値は `manual` です。

`enabled`: ブール型: `true` | `false`
activation-mode が `manual` の場合、enabled プロパティーは場所の現在の状態を反映します。このプロパティーは読み取り専用です。`netadm(1M)` を使って場所を有効または無効にすることで間接的に変更されます。

デフォルト値は `false` です。

`conditions`: 文字列のリスト: 条件式
`activation-mode` が `conditional-all` または `conditional-any` に設定されている場合、このプロパティーは、この場所をアクティブ化すべきかどうかを判断するテストを指定します。条件式は、「advertised-domain is sun.com」や「ip:bge0 is-not active」など、ブール値を割り当てることができる一連の条件から構成されます。これらの式の形式は、後述の「条件式」節で定義します。複数の条件が指定された場合、アクティブ化要件を満たすにはすべての条件が true である必要があるか (`activation-mode` が `conditional-all`)、いずれか 1 つの条件が true であればよいか (`activation-mode` が `conditional-any`) のどちらかです。

advertised-domain と system-domain の間には相違点があります。通知されたドメインは、DHCP サーバーによって通知された、DNSdmain や NISdmain などの外部通信を使用して学習されます。この属性は場所の条件によるアクティブ化に役立ちます。たとえば、通知されたドメインが mycompany.com である場合、「work」という場所をアクティブ化します。システムドメインとは、システムに現在割り当てられているドメインです。つまり、domainname(1M) コマンドで返される値です。この属性は、場所がアクティブ化されており、その特定のドメインに対してシステムが構成されてはじめて true になるので、ENM の条件によるアクティブ化に役立ちます。

default-domain: 文字列: システムドメイン名
システムに適用すべきドメイン名。このドメイン名は、NIS および LDAP で使用されます。

nameservices: 列挙値リスト: files | dns | nis | ldap
DNS、NIS、LDAP など、構成すべきネームサービスを指定します。

nameservices-config-file: 文字列: nsswitch.conf ファイルへのパス
使用する nsswitch.conf ファイルを指定します。このプロパティーには常に値が必要です。nameservices プロパティーで単一のネームサービスを指定した場合、このプロパティーには、デフォルトで、nameservices プロパティーで指定されたネームサービスの /etc/nsswitch.*nameservice* が含まれます。この値はユーザーが変更できます。nameservices プロパティーが複数のネームサービスを識別する場合、ユーザーは、このプロパティーに対して追加の値を指定する必要があります。

dns-nameservice-configsrc: 列挙値リスト: manual | dhcp
DNS ネームサービスの構成情報を取得するために使用すべきソースを指定します。manual が含まれる場合、残りの dns-nameservice-* プロパティーが使用されます。

dns-nameservice-domain: 文字列: ドメイン名
dns-nameservice-servers: 文字列リスト: ネームサーバーアドレス
dns-nameservice-search: 文字列リスト: ドメイン検索文字列
dns-nameservice-sortlist: 文字列リスト: *IP* アドレス、ネットマスクのペア
dns-nameservice-options: 文字列リスト: 変更されるリゾルバ変数
DNS が、構成されたネームサービスのいずれかであり、manual がその構成ソースの 1 つである場合、これらのプロパティーはその構成パラメータを指定します。servers プロパティーのみが必要です。search と domain は相互に排他的であり、一方しか指定できません。dhcp も使用されている場合、この値は DHCP サーバーが指定するドメインを上書きします。これらの値の形式は、resolv.conf(4) のそれぞれのオプションと同じです。

nis-nameservice-configsrc: 列挙値リスト: manual | dhcp
NIS ネームサービスの構成情報を取得するために使用すべきソースを指定します。manual が含まれる場合、残りの nis-nameservice-* プロパティーが使用されます。

`nis-nameservice-servers`: 文字列リスト: ネームサーバーアドレス
NIS が、構成されたネームサービスのいずれかであり、`manual` がその構成ソースの 1 つである場合、このプロパティーはそのサーバーアドレスを指定します。このプロパティーが指定されない場合、NIS クライアントはブロードキャストモードで起動します。

`ldap-nameservice-configsrc`: 列挙値リスト: `manual`
LDAP ネームサービスの構成情報を取得するために使用すべきソースを指定します。`manual` は、現在、LDAP の唯一のオプションです。したがって、LDAP 構成を完成させるには、`ldap-nameservice-*` プロパティーを指定する必要があります。

`ldap-nameservice-servers`: 文字列リスト: ネームサーバーアドレス
LDAP が、構成されたネームサービスのいずれかである場合、このプロパティーはそのサーバーアドレスを指定します。このプロパティーは必須であり、指定したサーバーには、クライアント構成を完成させるために使用されるクライアントプロファイルがあるものとみなされます。

`nfsv4-domain`: 文字列: *NVSv4* に使用するドメイン名
使用する NVSv4 ドメインを指定します。この値が指定されていない場合は、ネームサービスドメインが使用されます。

`ipfilter-config-file`: 文字列: *ipfilter IPv4* 構成ファイルへのパス
`ipfilter-v6-config-file`: 文字列: *ipfilter IPv6* へのパス
`ipnat-config-file`: 文字列: `ipnat` 構成ファイルへのパス
`ippool-config-file`: 文字列: `ippool` 構成ファイルへのパス
これらのプロパティーはそれぞれ、`ipfilter(5)` 構成のコンポーネントに対して、規則ファイルへのパスを指定します。特定の `config-file` プロパティーが設定された場合、指定されたファイルから構成が読み取られ、対応する IP フィルタのコンポーネントが有効になります。

`ike-config-file`: 文字列: *IKE* 構成ファイルへのパス
IKE 構成ファイルを指定します。値が指定された場合、指定されたファイルから構成が読み取られ、`svc:/network/ipsec/ike` サービスが有効になります。

`ipsecpolicy-config-file`: 文字列: *IPsec* ポリシー構成ファイルへのパス
IPsec ポリシー構成ファイルを指定します。値が指定された場合、指定されたファイルから構成が読み取られ、`svc:/network/ipsec/policy` サービスが有効になります。

ENM のプロパティー

`activation-mode`: 列挙値: `manual` | `conditional-all` | `conditional-any`
この ENM の自動アクティブ化のトリガーのタイプ。

デフォルト値は `manual` です。

`enabled`: ブール型: `true` | `false`
activation-mode が `manual` の場合、enabled プロパティーは ENM の現在の状態を反映します。このプロパティーは読み取り専用です。`netadm(1M)` を使って ENM を有効または無効にすることで間接的に変更されます。

デフォルト値は false です。

conditions: 文字列のリスト: 条件式
activation-mode が conditional-all または conditional-any に設定されている場合、このプロパティーは、この ENM をアクティブ化すべきかどうかを判断するテストを指定します。条件式は、「system-domain is sun.com」や「ip:bge0 is-not active」など、ブール値を割り当てることができる一連の条件から構成されます。これらの式の形式は、後述の「条件式」節で定義します。複数の条件が指定された場合、アクティブ化要件を満たすにはすべての条件が true である必要があるか (activation-mode が conditional-all)、いずれか 1 つの条件が true であればよいか (activation-mode が conditional-any) のどちらかです。

advertised-domain と system-domain の間には相違点があります。通知されたドメインは、DHCP サーバーによって通知された、DNSdmain や NISdmain などの外部通信を使用して学習されます。この属性は場所の条件によるアクティブ化に役立ちます。たとえば、通知されたドメインが mycompany.com である場合、「work」という場所をアクティブ化します。システムドメインとは、システムに現在割り当てられているドメインです。つまり、|domainname|(1M) コマンドで返される値です。この属性は、場所がアクティブ化されており、その特定のドメインに対してシステムが構成されてはじめて true になるので、ENM の条件によるアクティブ化に役立ちます。

fmri: 文字列: サービス *FMRI*
この ENM が SMF サービスとして実装された場合、このプロパティーはそのサービスを識別します。このプロパティーが指定された場合、ENM は、サービスを有効にすることによってアクティブ化され、サービスを無効にすることによって非アクティブ化されます。

start: 文字列: 起動コマンド
この ENM が SMF サービスとして実装されない場合、このプロパティーは、ENM を起動またはアクティブ化するために exec で実行すべきコマンドを識別します。FMRI プロパティーが設定されている場合、このプロパティーは無視されます。

stop: 文字列: 停止コマンド
この ENM が SMF サービスとして実装されない場合、このプロパティーは、ENM を停止または非アクティブ化するために exec で実行すべきコマンドを識別します。FMRI プロパティーが設定されている場合、このプロパティーは無視されます。

既知の WLAN のプロパティー

bssids: 文字列のリスト: WLAN BSSID (AP MAC アドレス)
特定のアクセスポイントを同じ名前/ESSID のほかのアクセスポイントより優先すべき場合は、このプロパティーを使用して AP の BSSID を指定できます。

priority: uint64: 優先順位の数値
この WLAN の相対的な優先順位。数値が小さいほど、高い優先度を表します。

その後、既知の WLAN オブジェクトセットを変更したときに、この数値の変更が必要になった場合は、変更されることがあります。たとえば、システムが wlanA と wlanB の 2 つの既知の WLAN オブジェクトで構成されているとします。wlanA には優先順位 1 が、wlanB には優先順位 2 が割り当てられています。新しい既知の WLAN、wlanC が作成され、優先順位 2 が割り当てられます。この場合、wlanA に優先順位 1、wlanC に優先順位 2、wlanB に優先順位 3 が割り当てられるように、リスト全体が更新されます。2 つの WLAN に同じ優先順位の値を割り当てることはできないので、優先順位 2 で wlanC を追加すると、wlanB の優先順位は強制的に下位に移動します。それ以外の WLAN が wlanB よりも低い優先順位にあった場合は、それに応じて移動されます。

keyslot: uint64: この WLAN に使用されるスロット
複数のキースロットをサポートする暗号化モードを WLAN で使用した場合、キーを配置するスロットは、このプロパティーで指定できます。指定しない場合、デフォルトでスロット 1 が使用されます。

keyname: 文字列のリスト: 安全なオブジェクト名
ユーザーが、dladm(1M) を使用して作成された安全なオブジェクトを既知の WLAN に関連付けられるようにします。

条件式

場所と ENM は、ユーザー指定の条件セットに基づいてアクティブ化できます。これらの条件式の構文を次の表にまとめます。

オブジェクトタイプ	オブジェクト	条件
ncp|ncu|enm|loc	name	is/is-not active

```
Object Type       Condition                          Object
-------------------------------------------------------------
essid             is/is-not/contains/does-not-contain name string
bssid             is/is-not                           bssid string
ip-address        is/is-not                           IPv4 or IPv6
                                                      address
ip-address        is-in-range/is-not-in-range         IPv4 or IPv6
                                                      address plus
                                                      netmask/prefixlen
advertised-domain is/is-not/contains/does-not-contain name string
system-domain     is/is-not/contains/does-not-contain name string
```

オプション

次のオプションは、特定の netcfg サブコマンドに関連付けられていません。

-d
コマンドファイルからサブコマンドを読み取る前に、すべての構成を削除します (次のオプションを参照)。

-f *command_file*
command_file から netcfg サブコマンドを読み取り、実行します。

サブコマンド

次のサブコマンドがサポートされています。

`cancel`
永続ストレージに対する現在の変更を確定することなく、現在のプロファイル指定を終了し、次の上位のスコープに移ります。

このサブコマンドは、NCP およびプロファイルのスコープで有効です。

`clear` *prop-name*
指定したプロパティーの値を消去します。

このサブコマンドはプロファイルのスコープで有効です。

`commit`
永続ストレージに現在のプロファイルを確定します。構成が正しく設定されていないと確定できないので、この操作では、オブジェクトでの検証も自動的に行われます。確定操作は、現在のスコープから外れるときに自動的に試行されます (`end` サブコマンドまたは `exit` サブコマンドの使用時)。

このサブコマンドはプロファイルのスコープで有効です。

非対話型モードでは、値を変更するすべてのサブコマンドで暗黙に確定が行われるので、確定操作は必要ありません。

`create` [`-t` *template*] *object-type* [*class*] *object-name*
指定されたタイプと名前のメモリー内プロファイルを作成します。`-t` *template* オプションは、新しいオブジェクトが、*template* と同一であるべきことを指定します。ここで、*template* は同じタイプの既存のオブジェクトの名前です。`-t` オプションを指定しない場合、新しいオブジェクトはデフォルト値で作成されます。NCP の場合、「User」NCP のみを作成できます。

このサブコマンドは、グローバルおよび NCP のスコープで有効です。

`destroy` { `-a` | *object-type* [*class*] *object-name* }
メモリーおよび永続ストレージから、すべてまたは指定したプロファイルを削除します。この処理は即座に実行されます。確定する必要はありません。破棄したオブジェクトを元に戻すことはできません。

特定のオブジェクトが指定された場合、このサブコマンドは、グローバルおよび NCP のスコープで有効です。`-a` オプションはグローバルスコープでのみ有効です。

`end`
現在のプロファイル指定を終了し、次の上位のスコープに移ります。終了する前に、現在のオブジェクトが検証され確定されます。検証操作または確定処理のどちらかが失敗した場合、該当するエラーメッセージが発行され、ユーザーには、現在の変更を確定せずに終了するか、現在のスコープのまま編集を続行するかを選択する機会が与えられます。

このサブコマンドはどのスコープでも有効です。

`exit`
: `netcfg` セッションを終了します。編集前に、現在のプロファイルが検証され確定されます。どちらかの処理が失敗した場合、該当するエラーメッセージが発行され、ユーザーには、現在の変更を確定せずに終了するか、現在のスコープのまま編集を続行するかを選択する機会が与えられます。

  このサブコマンドはどのスコープでも有効です。

`export` [`-d`] [`-f` *output-file*] [*object-type* [*class*] *object-name*]
: 現在または指定したスコープの現在の構成を、標準出力または `-f` オプションで指定したファイルに出力します。`-d` オプションは、出力の最初の行として `destroy -a` を生成します。このサブコマンドを使用すると、コマンドファイルに適した形式で出力が生成されます。Automatic NCP および Automatic、NoNet、ユーザー場所などのシステム作成オブジェクトは、エクスポートできません。

  このサブコマンドはどのスコープでも有効です。

`get` [`-V`] *prop-name*
: 指定したプロパティーの現在 (メモリー内) の値を取得します。デフォルトで、プロパティー名と値の両方が出力されます。`-V` オプションを指定した場合は、プロパティー値のみが表示されます。

  このサブコマンドはプロファイルのスコープで有効です。

`help` [*subcommand*]
: 一般ヘルプまたは特定のサブコマンドに関するヘルプを表示します。

  このサブコマンドはどのスコープでも有効です。

`list` [`-a`] [*object-type* [*class*] *object-name*]
: 現在または指定したスコープに存在するすべてのプロファイル、プロパティーと値のペア、およびリソースを一覧表示します。オブジェクトのプロパティーを一覧表示するときには、デフォルトの動作で、指定した構成に適用されるプロパティーのみを一覧表示します。つまり、`ipv4-addrsrc` が `dhcp` である IP NCU を一覧表示する場合、`ipv4-addr` プロパティーは一覧表示されません。`-a` オプションを含めると、現在の設定に適用されるかどうかにかかわらず、すべてのプロパティーが一覧表示されます。

  このサブコマンドはどのスコープでも有効です。

`revert`
: 現在のプロファイルに対する現在の変更を削除し、永続ストレージから値を元に戻します。

  このサブコマンドはプロファイルのスコープで有効です。

`select` *object-type* [*class*] *object-name*
現在のスコープのレベルで使用できるプロファイルのいずれかを選択し、オブジェクトのスコープまで下げます。選択したオブジェクトは、永続ストレージからメモリーに読み込まれます。

このサブコマンドは、グローバルおよびNCPのスコープで有効です。

`set` *prop-name=value1*[,*value2...*]
指定したプロパティーの現在(メモリー内)の値を設定します。非対話型モードで実行した場合、永続ストレージにも変更が確定されます。

複数の値を持つプロパティーの値の区切り文字は「,」(コンマ)です。このようなプロパティー内の個々の値にコンマが含まれる場合、コンマをエスケープする必要があります(つまり、\, と記述)。単一の値のみを持つプロパティー内のコンマは、区切り文字として解釈されず、エスケープする必要はありません。

このサブコマンドはプロファイルのスコープで有効です。

`verify`
現在のメモリー内オブジェクトに有効な構成があることを確認します。

このサブコマンドはプロファイルのスコープで有効です。

`walkprop` [`-a`]
現在のプロファイルに関連付けられた各プロパティーを調査します。プロパティーごとに、名前と現在値が表示され、ユーザーが現在の値を変更するためのプロンプトが表示されます。

複数の値を持つプロパティーの値の区切り文字は「,」(コンマ)です。このようなプロパティー内の個々の値にコンマが含まれる場合、コンマをエスケープする必要があります(つまり、\, と記述)。単一の値のみを持つプロパティー内のコンマは、区切り文字として解釈されず、エスケープする必要はありません。

デフォルトでは、すでに設定されているプロパティーに基づいて、必要なプロパティーのみが調査されます。つまり、`ipv4-addrsrc` が `dhcp` に設定されている場合、`ipv4-addr` は調査されません。`-a` オプションを含めると、使用可能なすべてのプロパティーが調査されます。

このサブコマンドはプロファイルのスコープで有効です。

このサブコマンドは、対話型モードでのみ意味があります。

使用例

例1 NCUのプロパティーを設定する

次のコマンドは、コマンド行から(つまり非対話型モードで)NCUのプロパティーを設定します。

```
# netcfg "select ncp User; select ncu phys net1; set link-mtu=1492"
```

例 2 最上位プロファイルを一覧表示する

次のコマンドは、コマンド行からすべての最上位プロファイルを一覧表示します。

```
# netcfg list
NCPs:
        Automatic
        User
Locations:
        Automatic
        home
        NoNet
        office
ENMs:
        enmtest
        myenm
WLANs:
        sunwifi
        coffeeshop
        linksys
```

例 3 場所プロファイルを破棄する

次のコマンドは、コマンド行から場所プロファイルを破棄します。

```
# netcfg destroy loc home
Destroyed loc 'home'
```

例 4 NCU プロファイルを作成する

次のコマンドシーケンスは、NCU プロファイルを対話形式で作成します。

```
# netcfg
netcfg> select ncp user
netcfg:ncp:User> create ncu ip net1
Created ncu 'net1'.  Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> static
ipv4-addr> 168.1.2.3
netcfg:ncp:User:ncu:net1> list
ncu:net1
        type:             interface
        class:            ip
        parent:           "User"
        enabled           true
        ip version:       ipv4
        ipv4-addrsrc:     static
        ipv4-addr:        "168.1.2.3"
        ipv6-addrsrc      dhcp,autoconf
netcfg:ncp:User:ncu:net1> commit
Committed changes
```

例 4　NCU プロファイルを作成する　　*(続き)*

```
netcfg:ncp:User:ncu:net1> end
netcfg:ncp:User> exit
```

例 5　ENM を操作する

次のコマンドシーケンスは、既存の ENM を選択し、その内容を表示して、プロパティー値を変更します。

```
# netcfg
netcfg>select enm myenm
netcfg:enm:myenm>list
ENM:myenm
        activation-mode manual
        enabled         true
        start           "/usr/local/bin/myenm start"
        stop            "/usr/local/bin/myenm stop"
netcfg:enm:myenm>set stop="/bin/alt_stop"
netcfg:enm:myenm>list
ENM:myenm
        activation-mode manual
        enabled         true
        start           "/usr/local/bin/myenm start"
        stop            "/bin/alt_stop"
netcfg:enm:myenm>exit
Committed changes
```

例 6　静的アドレスを構成する

次のコマンドシーケンスは、Home NCP を使用して、インタフェース `bge0` で、192.168.2.12/24 の静的アドレスを構成します。このインタフェースは、Home NCP がアクティブ化されているときに、デフォルトで有効になります。

```
# netcfg
netcfg> create ncp Home
```

最初に、phys NCU を構成します。

```
netcfg:ncp:Home> create ncu phys bge0
Created ncu 'bge0'.  Walking properties ...
activation-mode (manual) [manual|prioritized]>
link-mac-addr>
link-autopush>
link-mtu>
netcfg:ncp:Home:ncu:bge0> end
Committed changes
netcfg:ncp:Home>
```

次に、IP NCU を構成します。

例 6　静的アドレスを構成する　　(続き)

```
netcfg:ncp:Home> create ncu ip bge0
Created ncu 'bge0'.  Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]>
ipv4-addrsrc (dhcp) [dhcp|static]> static
ipv4-addr> 192.168.2.10/24
ipv4-default-route>
ipv6-addrsrc (dhcp,autoconf) [dhcp|autoconf|static]>
ipv6-default-route>
netcfg:ncp:Home:ncu:bge0> list
ncu:bge0
    type                interface
    class               ip
    parent              "Home"
    enabled             true
    ip-version          ipv4,ipv6
    ipv4-addrsrc        static
    ipv4-addr           "192.168.2.10/24"
    ipv6-addrsrc        dhcp,autoconf
netcfg:ncp:Home:ncu:bge0> exit
Committed changes
#
```

netadm(1M) を使用して Home NCP に切り替えます。

例 7　条件に基づいて場所を構成する

次のコマンドシーケンスは、条件に基づいて場所を構成します。場所は、IP アドレスが 10.0.8.0/24 サブネット内にあるときはいつでもアクティブ化されます。また、NIS がこの場所で構成されます。

```
netcfg> select loc office
netcfg:loc:office> list
loc:office
       activation-mode                conditional-any
       conditions                     "ip-address is 10.0.8.0/24"
       enabled                        false
       nameservices                   dns,nis
       nameservices-config-file       "/etc/nsswitch.nis"
       dns-nameservice-configsrc      dhcp
       nis-nameservice-configsrc      manual
       nis-nameservice-servers        "10.2.18.24"
       default-domain                 "labs.east.sun.com"
netcfg:loc:office>
```

例 8 既知の WLAN を作成する

次のコマンドシーケンスは、WEP キーを持つ coffeeshop という既知の WLAN を確立します。

```
netcfg> select wlan coffeeshop
netcfg:wlan:coffeeshop> list
known wlan:coffeeshop
       priority                2
       keyname                 "foo"
       security-mode           wep
netcfg:wlan:coffeeshop> set priority=1
netcfg:wlan:coffeeshop> end
Committed changes
```

例 9 現在の構成をファイルにエクスポートする

次のコマンドは現在の構成をファイルにエクスポートします。

```
netcfg> export -f /tmp/nwam.config
```

または、UNIX コマンド行から同じタスクを実行します。

```
# netcfg export -f /tmp/nwam.config
```

例 10 現在の構成をファイルからインポートする

次のコマンドは現在の構成をファイルからインポートします。

```
# netcfg -f /tmp/nwam.config
```

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

dladm(1M), domainname(1M), netadm(1M), netcfgd(1M), nwamd(1M), resolv.conf(4), attributes(5), ipfilter(5)

JDS/GNOME マニュアルページコレクションで利用可能な nwam-manager(1M) も参照してください。

名前

netcfgd – ネットワーク構成管理デーモン

形式

/lib/inet/netcfgd

機能説明

netcfgd は、ネットワーク構成を管理するシステムデーモンです。このデーモンは、svc:/network/netcfg:default により自動的に起動されるので、直接起動しないようにしてください。これは、プログラミングインタフェースの構成要素ではありません。

netcfgd は、ネットワーク構成リポジトリへのアクセスを管理し、リーダーとライターが適切な承認を保持するようにします。具体的には、ネットワーク構成の読み取りを実行するアプリケーションは solaris.network.autoconf.read 承認を保持している必要があり、ネットワーク構成の書き込みを行うアプリケーションは solaris.network.autoconf.write 承認を保持している必要があります。これらの承認は auth_attr(4) に登録され、prof_attr(4) 内の Network Autoconf プロファイルに含められます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

svcs(1), netcfg(1M), nwamd(1M), svcadm(1M), auth_attr(4), prof_attr(4), attributes(5), smf(5)

JDS/GNOME マニュアルページコレクションで利用可能な nwam-manager(1M) も参照してください。

名前

netservices – ネットワークサービスの有効化または無効化

形式

```
netservices open

netservices limited
```

機能説明

netservices コマンドは、Solaris サービス管理機能 smf(5) を使用して、ネットワーク経由でリモートクライアントから受け入れるサービスを制御します。

limited コマンド行引数を指定して netservices を呼び出すと、SSH デーモン sshd(1M) を除くすべてのネットワークサービスが無効化されるか、ローカルの要求への応答のみに制限されます。

open コマンド行引数を設定して netservices を呼び出すと、以前の Solaris リリースのように、多数のネットワークサービスが有効になります。

netservices により有効になる構成セットをカスタマイズするには、svcadm(1M) を使用してサービスを個別に有効または無効にします。リモートクライアントからの入力をサービスが受け入れるかどうかを決定するプロパティーの設定には、svccfg(1M) を使用してください。サービスインスタンスの名前およびそのプロパティーについては、各サービスのマニュアルページを参照してください。

netservices コマンドは、インタフェースの安定性が「廃止」であることに留意してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	廃止

関連項目

svcadm(1M), svccfg(1M), sshd(1M), attributes(5), smf(5)

名前

netstat – ネットワークのステータスを表示

形式

```
netstat [-anvR] [-f address_family] [-P protocol]

netstat -g [-nv] [-f address_family]

netstat -p [-n] [-f address_family]

netstat -s [-f address_family] [-P protocol]
     [-T u | d ] [interval [count]]

netstat -m [-T u | d ] [-v] [interval [count]]

netstat -i [-I interface] [-an] [-f address_family]
     [-T u | d ] [interval [count]]

netstat -r [-anvR] [-f address_family | filter]

netstat -M [-ns] [-f address_family]

netstat -D [-I interface] [-f address_family]

netstat -d [-f address_family]
```

機能説明

netstat コマンドは、ネットワーク関連のデータ構造の内容を、指定されたオプションに従って、さまざまな形式で表示します。

netstat コマンドの形式は上のセクションに示されていますが、次のようないくつかの形態があります。

- 1つめの形式は、必ずしも引数を指定する必要がなく、それぞれのプロトコルについてアクティブなソケットを示します。
- 2 番目、3 番目、4 番目の形式 (-g、 -p、および -s オプション) は、さまざまなネットワークデータ構造から情報を取り出して表示します。
- 5 番目の形式は (-m オプション) STREAMS メモリーの統計情報を表示します。
- 6 番目の形式は (-i オプション) インタフェースの状態を表示します。
- 7 番目の形式 (-r オプション) はルーティングテーブルを表示します。
- 8 番目の形式は (-M オプション) マルチキャストルーティングテーブルを表示します。
- 9 番目の形式 (-D オプション) は 1 つまたはすべてのインタフェースの DHCP の状態を表示します。
- 10 番目の形式 (-d オプション) は、宛先キャッシュエントリのテーブルを表示します。

これらの形式について、以下により詳しく説明します。

引数を 1 つも指定しない場合 (最初の形式), netstat は f オプションで設定を変更しないかぎり、PF_INET、 PF_INET6、 および -PF_UNIX の接続ソケットを表示します。

オプション

`-a`

すべてのソケット、 すべてのルーティングテーブルエントリ、すべての物理および論理インタフェースの状態を表示します。通常は、サーバープロセスに使用されるリスナーソケットは表示されません。ほとんどの場合、インタフェース、ホスト、ネットワーク、およびデフォルトルートだけが表示されます。また、インタフェースでステータスが表示されるのは、物理インタフェースのみです。

`-d`

宛先キャッシュエントリテーブルを表示します。下の「表示」を参照してください。

`-f` *address_family*

すべての表示を *address_family* で指定されたものだけに限定します。指定できる *address_family* の値は次のとおりです。

`inet` IPv4 情報を表示する `AF_INET` アドレスファミリ用

`inet6` IPv6 情報を表示する `AF_INET6` アドレスファミリ用

`unix` `AF_UNIX` アドレスファミリ用

`sdp` Socket Description Protocol (SDP) プロトコルおよびアドレスファミリの場合、 SDP ソケットに関して表示されるアドレス状態を次に示します。各状態の意味のあとに、`netstat` により表示されるフラグを示します。

`LST` Listen
`EST` Established
`PL` Path Lookup
`HS` Hello Request Sent
`HR` Hello Request Received
`HAR` Hello Ack Recvd
`HAS` Hello Ack sent
`DR` Fin received
`DS` Fin sent
`DSA` Fin Ack recvd
`DRC` Simultaneous Disconnect
`DSC` Disconnect sent (peer already closed)
`TW1` Time Wait 1
`TW2` Time Wait 2
`CLD` Closed
`ERR` Error
`INV` Invalid
`UNK` Unknown

SDP プロトコルおよびアドレスファミリの場合、netstat は次の列ヘッダーを表示します。

ローカルアドレス	ローカル IP アドレス
リモートアドレス	リモート IP アドレス
状態	ソケットの現在の状態
RxBPending	未読バイト
TxBQueued	Tx (TxBPosted を含む) のキューに入っているバイト数
TxBPosted	転送用に HW に送信済みのバイト数
LAdvtSz	ローカルの通知済みバッファーサイズ
RAdvtSz	リモートの通知済みバッファーサイズ
LAdvtBuff	ローカルの通知済み Rx バッファー数
RAdvtBuff	リモートの通知済み Rx バッファー数
LPostBuff	現在の送信済み Rx バッファー数

-f *filter*

-r オプションとともに使用した場合のみ、経路の表示を指定されたフィルタにマッチするものに限定します。フィルタのルールは *keyword:value* のペアで構成されます。既知の keyword と value の構文は次のとおりです。

af:{inet|inet6|unix|sdp|*number*}	アドレスファミリを選択します。これは -f *address_family* と同じで、どちらの構文もサポートされます。
outif:{*name*|*ifIndex*|any|none}	出力インタフェースを選択します。名前 (hme0 など) または ifIndex 番号 (たとえば 2) でインタフェースを指定することができます。any が指定された場合は、指定されたインタフェースを持つすべての経路がフィルタにマッチします (null 以外のすべて)。 が指定された場 合は、 null インタフェースを持つすべての経路がマッチします。インタフェースのインデックス番号 (*ifIndex*) は ifconfig(1M) の -a オプションで確認できます。
dst:{*ip-address*[/*mask*]|any|none}	着信先 IP アドレスを選択します。マスク長とともに指定した場合は、長さが同じか、またはより長い (より限定された) マスクにマッチする経路が選択されます。any が指定された場

	合は、0 以外のすべてのアドレスが選択されます。none が指定された場合は、アドレス 0 が選択されます。
flags:[+ -]?[ABDGHLMSU]+	指定されたフラグでタグ付けされた経路を選択します。デフォルトでは、フラグが指定されたとおりに設定されている場合にのみマッチします。先頭に + を付けることにより、指定されたフラグが設定されている場合にのみマッチするようになります。先頭に - を付けることにより、指定されたフラグが設定されていない場合にのみマッチするようになります。

-f オプションを複数使用することにより、複数のフィルタを指定することができます。例:

```
% netstat -nr -f outif:hme0 -f outif:hme1 -f dst:10.0.0.0/8
```

上のコマンドでは、マスク長が8以上でネットワーク 10.0.0.0/8 に属する経路と、出力インタフェースが hme0 または hme1 である経路が表示され、それ以外の経路は全て除外されます。

-g
すべてのインタフェースのマルチキャストグループメンバーシップを表示します。-v オプションが含まれている場合は、ソースに特定のメンバーシップ情報も表示されます。下の「表示」を参照してください。

-i
IP トラフィック用に使用されるインタフェースの状態を表示し ます。通常は、物理インタフェースの統計情報を表示します。-a オプションと組み合わせて使用された場合は、論理インタフェースの情報も報告されます。ifconfig(1M) を参照してください。

-m
STREAMS メモリーの統計情報を表示します。

-n
ネットワークアドレスを数字で表示します。netstat は、通常はアドレスをシンボルで表示します。このオプションは、どの表示形式でも使用できます。

-p
Net to Media テーブルを表示します。下の「表示」を参照してください。

-r
ルーティングテーブルを表示します。通常は、インタフェース、ホスト、ネットワーク、およびデフォルトルートだけが表示されますが、このオプションと -a オプションを組み合わせることにより、キャッシュを含むすべての経路が表示さ

れます。マルチキャストルートを設定していない場合は、カーネルが必要に応じてこの種のエントリを取得しても、-ra によりマルチキャストルーティングエントリが表示されないことがあります。

-s
プロトコルごとに統計情報を表示します。-M オプションとともに使用した場合は、この代わりにマルチキャストルーティング統計情報を表示します。-a オプションとともに使用した場合は、システム全体の統計情報に加えて、インタフェースごとの統計情報が表示されます。下の「表示」を参照してください。

-T u|d
タイムスタンプを表示します。

時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。標準の日付フォーマットに d を指定します。date(1) を参照してください。

-v
冗長。ソケット、STREAMS メモリー統計情報、ルーティングテーブル、マルチキャストグループメンバーシップの追加情報を表示します。

-I *interface*
特定のインタフェースの状態を表示します。*interface* は hme0 や eri0 などの有効なインタフェース名です。通常は物理インタフェースのステータスと統計情報が表示されます。このオプションを -a オプションとともに使用した場合、論理インタフェースの統計情報も報告されます。

-M
マルチキャストルーティングテーブルを表示します。-s オプションとともに使用した場合、この代わりにマルチキャストルーティングの統計情報を表示します。

-P *protocol*
表示するすべてのソケットの状態や統計情報を *protocol* に関連するものに限定します。指定できるプロトコルは ip, ipv6, icmp, icmpv6, icmp, icmpv6, igmp, udp, tcp, rawip のいずれかです。rawip を raw として指定することもできます。コマンドはプロトコルオプションを小文字でのみ受け入れます。

-D
DHCP で構成されたインタフェースのステータスを表示します。

-R
この修飾子を指定すると、ソケットとルーティングテーブルエントリの拡張セキュリティ属性が表示されます。-R 修飾子はシステムが Solaris Trusted Extensions 機能とともに構成されている場合にのみ利用可能です。

-r オプションのみの場合、このオプションはルーティングエントリのゲートウェイセキュリティ属性だけを表示します。 セキュリティ属性について詳しくはroute(1M) を参照してください。

最初の形式のコマンドを使用してソケット情報を表示する際に、このオプションはマルチレベルポート(MLP) ソケットの追加情報を表示します。これには次のようなものが含まれます。

- ソケットが接続されている場合、ピアのラベル。
- 次のフラグがソケットの "State" 出力に追加されます。

  P　ソケットはゾーンプライベート IP アドレス上の MLP です。

  S　ソケットはゾーン間で共有される IP アドレス上の MLP です。

オペランド

interval　*interval* 秒ごとに前回の表示から積算した統計情報を表示し、*count* が指定されていない場合は永久に繰り返します。*interval* とともにブートされた場合、netstat の出力の最初の行には、最後のリブートから積算された統計情報が表示されます。

次のオプションは *interval* をサポートします: `-i`, `-m`, `-s` および `-Ms`. いくつかの値は構成パラメータで、単にインターバルごとに表示されます。

count　*interval* で間隔が指定されている際に *count* で指定された回数だけ、インタフェースの統計情報の表示を繰り返します。

表示

アクティブなソケット (第 1 の形式)

それぞれのアクティブなソケットの情報には、ローカルとリモートのアドレス、送信と受信の待ち行列サイズ (バイト数)、送信および受信ウィンドウ (バイト数) 、プロトコルの内部状態があります。

ソケットアドレスの表示に使用される記号形式は通常、次のどちらかです。

`hostname.`*port*

上記はソケットアドレスがホスト名を表す場合です。

network`.`*port*

上記はソケットアドレスが特定のホストではなくネットワークを表す場合です。

ソケットに関連する数値ホストアドレスやネットワーク番号を使って、それに対応する記号ホスト名やネットワーク名が *hosts* や *networks* データベースから検索されます。

アドレスのネットワークやホスト名が不明な場合、または `-n` オプションが指定されている場合、ネットワークアドレスが数値で表示されます。無指定または「ワイルドカード」のアドレスやポートは 「`*`」 で表示されます。インターネットの命名規則については `inet(7P)` と `inet6(7P)` を参照してください。

SCTP ソケットについては、エンドポイントが複数のアドレスで表されることがあるため、冗長オプション (-v) はすべてのローカルとリモートのアドレスを表示します。

TCP ソケット

TCP ソケットの状態を表す値には、次のものがあります。

BOUND	バインドされました。接続または待機の準備ができている。
CLOSED	クローズされました。ソケットは使用されていない。
CLOSING	クローズされ、リモートシャットダウンが行われた。肯定応答を待っている。
CLOSE_WAIT	リモートシャットダウン。ソケットのクローズを待っている。
ESTABLISHED	接続が確立されました。
FIN_WAIT_1	ソケットがクローズされました。接続を停止している。
FIN_WAIT_2	ソケットがクローズされました。リモートからのシャットダウンを待っている。
IDLE	アイドル。オープンされているが、バインドされていない。
LAST_ACK	リモートシャットダウンの後、クローズされました。肯定応答を待っている。
LISTEN	着信する接続を待機している。
SYN_RECEIVED	進行中の接続の最初の同期化をはかる。
SYN_SENT	アクティブに接続を確立しようとしている。
TIME_WAIT	クローズの後、リモートシャットダウンの再送信を待っている。

SCTP ソケット

SCTP ソケットの状態を表す値には、次のものがあります。

CLOSED	クローズされました。ソケットは使用されていない。
LISTEN	着信するアソシエーションを待機している。
ESTABLISHED	アソシエーションが確立されました。
COOKIE_WAIT	ピアに INIT が送信されました。肯定応答を待っている。
COOKIE_ECHOED	INIT-ACK からのクッキーがピアに送信されました。肯定応答を待っている。
SHUTDOWN_PENDING	上位レイヤーから SHUTDOWN を受信しました。ピアからのすべての未処理 DATA の肯定応答を待っている。
SHUTDOWN_SENT	すべての未処理データが SHUTDOWN_SENT 状態で肯定されました。ピアに SHUTDOWN が送信されました。肯定応答を待っている。

SHUTDOWN_RECEIVED	ピアから SHUTDOWN を受信しました。すべての未処理の DATA の肯定応答を待っている。
SHUTDOWN_ACK_SENT	すべての未処理データが SHUTDOWN_RECEIVED 状態で肯定されました。ピアに SHUTDOWN_ACK が送信されました。

ネットワークデータ構造 (第 2 - 第 5の形式)

表示形式次は、 -g, -m, -p, -s オプションのどれを指定するかによって異なります。

-g　マルチキャストグループメンバーシップのリストを表示します。

-m　メモリーの使用量を表示します (たとえば、STREAMS mblks) 。

-p　ネットワークとメディアのマッピングテーブルを表示します。IPv4 の場合は、アドレス解決テーブルが表示されます。arp(1M) を参照してください。IPv6 の場合は、近隣キャッシュが表示 されます。

-s　さまざまなプロトコルレイヤの統計を表示します。

統計では、MIB 指定変数が使用されます。ipForwarding の値は次のように定義されています。

forwarding(1)	ゲートウェイとして機能している。
not-forwarding(2)	ゲートウェイとして機能していない。

IPv6 および ICMPv6 プロトコルレイヤはインタフェースごとの統計を保持します。-a オプションが -s オプションと共に指定された場合は、合計された値とともにインタフェースごとの統計が表示されます。そうでない場合は、合計の統計情報だけが表示されます。

第 2、第 3、第 4 の形式のコマンドでは、すくなくとも -g, -p, -s のいずれかを指定する必要があります。これらのオプションは、どのような組み合わせで指定することもできます。-m (第 5 の形式) オプションは -g, -p, -s オプションのいずれの組み合わせとも同時に指定することができます。これらのオプションを複数指定した場合、netstat はそれぞれに対応した情報を表示します。

インタフェースステータス (第 6 の形式)

インタフェースステータスでは、現在のすべてのインタフェースの情報が 1 行に 1 つずつ表示されます。-I オプションでインタフェースを指定すると、そのインタフェースの情報だけが表示されます。

この情報には、インタフェース名、mtu (最大伝送単位、または最大パケットサイズ) (ifconfig(1M) を参照)、インタフェースが接続されているネットワーク、各インタフェースのアドレス、インタフェースに対応するカウンタが含まれます。カウンタの種類には、入力パケット数、入力エラー数、出力パケット数、出力エラー数、衝突数のカウンタがあります。二地点間インタフェースの場合、Net および Dest フィールドはリンクの相手の名前またはアドレスです。

-a オプションが -i オプションまたは -I オプションのいずれかと同時に指定された場合、出力に物理インタフェースの名前が含まれ、それぞれの論理インタフェースの入力パケット数と出力パケット数が追加の情報とともに表示されます。

-n オプションを指定すると、インタフェース名の代わりに IP アドレスが表示されます。

オプションの *interval* を指定すると、出力は、ユーザーが中断するか、または *count* に到達するまで *interval* 秒ごとに表示されます。「オペランド」の節を参照のこと。

物理インタフェースを指定する場合は -I オプションを使用します。*interval* オペランドとともに使用した場合は、-I オプションのための表示は次のような形式になります。

```
input    eri0          output        input          (Total)   output
packets  errs  packets errs  colls   packets  errs  packets  errs   colls
227681   0     659471  1     502     261331   0     99597    1      502
10       0     0       0     0       10       0     0        0      0
8        0     0       0     0       8        0     0        0      0
10       0     2       0     0       10       0     2        0      0
```

入力インタフェースが指定されていない場合は、アドレスファミリ inet または inet6 の最初のインタフェースが表示されます。

ルーティングテーブル (第 7 の形式)

ルーティングテーブルには、使用可能な経路とそれぞれのステータスが表示されます。各経路には、宛先のホストまたはネットワークと、パケットの転送で使用するゲートウェイが表示されます。*flags* 列には経路のステータスが表示されます。表示に使用されるフラグの意味は次のとおりです。

U　経路が up (オン) である。

G　経路がゲートウェイに向かっている。

H　経路がネットワークではなくホストに向かっている。

M　-multirt オプションによって確立された冗長経路である。

S　-setsrc オプションによって確立された経路である。

D　リダイレクトによって動的に作成された経路である。

B　パケットは通知なく削除される (RTF_BLACKHOLE が設定されている)。

R　パケットの削除時に ICMP エラーが送信される (RTF_REJECT が設定されている)。

I　-indirect オプションを使用して間接経路 (ゲートウェイへの直接到達は不可能) が確立されました。

Z (非大域の排他的 IP ゾーンのみ) 大域ゾーン内で zonecfg(1M) を使用して構成された経路指定情報に基づき、ブート時に経路が静的に追加されました。

-a オプションが指定された場合は、次のフラグを持つルーティングエントリも表示されます。

b ブロードキャストアドレス。

C オンリンク宛先のインタフェースホスト経路エントリのクローンを作成します。

L ホストのローカルアドレス。

インタフェース経路は、ローカルホストの接続されているインタフェースごとに作成されます。このようなエントリのゲートウェイフィールドには、発信インタフェースのアドレスが表示されます。

use 列には、その経路を使って送信または転送されたパケット数が表示されます。

interface エントリは、その経路に使用されたネットワークインタフェースを示します。

マルチキャストルーティングテーブル (第 8 の形式)

マルチキャストルーティングテーブルは、仮想インタフェーステーブルと実際のルーティングテーブルから構成されています。

インタフェース情報 (第 9 の形式)

DHCP インタフェース情報は、インタフェース名、その現在の状態、リース情報、パケット数、フラグのリストからなります。

表示の内容は RFC 2131 で提唱された仕様に従います。

リース情報には次の内容が含まれます。

- リース開始日時
- リース更新の開始日時
- リース期限の日時

現在定義されているフラグには次のようなものがあります。

BOOTP インタフェースが BOOTP から取得したリースを保持している (IPv4 のみ)。

BUSY インタフェースは DHCP トランザクションのためにビジー状態である。

PRIMARY このインタフェースは主インタ フェースである。dhcpinfo(1) および ifconfig(1M) を参照してください。

FAILED インタフェースはエラー状態であり、手動でリスタートする必要があります。

パケットカウントは、送信されたパケット、受信したパケット、DHCP クライアントによって拒否 (DECLINE) されたリース提案の数を保持します。3 つのカウンタは、リース取得時にすべて 0 に初期化され、その後インクリメントされます。そのインタフェースに対するリース更新期間の開始時にもカウンタはリセットされます。このように、カウンタの保持する送信、受信、拒否パケットの数は、現在のリースを取得する際、または、将来のリースを取得しようとしている場合のいずれかになります。

宛先キャッシュエントリテーブル (10 番目の形式)

宛先キャッシュエントリには、記録されたパス MTU、エントリの経過時間 (秒)、およびフラグが表示されます。P フラグは、パス MTU が記録されることを示します。S フラグは、パス MTU が IP により許可される最小値よりも小さいことを示します。U フラグは、一部のトランスポート基準値 (往復時間など) が宛先キャッシュエントリにキャッシュされることを示します。

ファイル

/etc/default/inet_type　DEFAULT_IP 設定

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

arp(1M), dhcpinfo(1), dhcpagent(1M), ifconfig(1M), iostat(1M), kstat(1M), mibiisa(1M), savecore(1M), vmstat(1M), zonecfg(1M), hosts(4), inet_type(4), networks(4), protocols(4), services(4), attributes(5), dhcp(5), kstat(7D), inet(7P), inet6(7P)

Droms, R., RFC 2131, Dynamic Host Configuration Protocol, Network Working Group, March 1997.

Droms, R. RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6). Cisco Systems. July 2003.

注意事項

インタフェースの情報を表示する際 に、netstat は/etc/default/inet_type の DEFAULT_IP 設定を優先します。この設定が IP_VERSION4 の場合、 netstat netstat は IPv6 に関するインタフェース、統計情報、接続、経路などの情報を表示しません。

しかし、コマンド行で /etc/default/inet_type のDEFAULT_IP 設定をオーバーライドすることができます。たとえば、コマンド行で inet6 アドレスファミリまたは IPv6 プロトコルを使用することで、明示的に IPv6 情報を要求することができ、これは DEFAULT_IP の設定より優先されます。

カーネルクラッシュの後でネットワークステータス情報を調べたい場 合は、mmdb(1) ユーティリティーを savecore(1M) 出力に対して使用します。

netstat ユーティリティーは /dev/tcp を開いてクエリを発行するこ とによって TCP 統計情報を取得します。このため、netstat が接続ステータスを報告する際に、使用されていない接続が IDLE ステータスで表示されることがあります。

以前のバージョンの netstat には kstat(7D) を使ってカーネル統計を報告する非公式の方法がありました。この機能は削除されています。kstat(1M) を使用してください。

netstat の出力は netstat が実行されているゾーンに関連する情報だけに制限されています。(これは、共有 IP と 占有 IP のどちらのゾーンにもあてはまります。)

名前

netstrategy – return network configuration information

形式

`/usr/sbin/netstrategy`

機能説明

The `netstrategy` command determines the network configuration strategy in use on a system and returns information in a form that is easily consumable by a script. The command returns three tokens:

<root filesystem type> *<primary interface>* *<network config strategy>*

These tokens are described as follows:

<root filesystem type>
: Type of filesystem that contains the bootable kernel, as would be specified in the *fstype* column of the `mnttab(4)`.

<primary interface>
: Name of the primary network interface. For a diskless machine, this is the interface used to load the kernel.

<network config strategy>
: The means by which a system obtains its IP address for booting. This can be one of `rarp`, `dhcp`, or `none`.

The `netstrategy` command is not intended for use on a command line.

オプション

The `netstrategy` command has no options.

終了ステータス

`0` Success.

`!=0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`ifconfig(1M)`, `mnttab(4)`, `attributes(5)`

名前

newaliases – rebuild the data base for the mail aliases file

形式

`newaliases`

機能説明

`newaliases` rebuilds the random access data base for the mail aliases file `/etc/mail/aliases`.

`newaliases` accepts all the flags that `sendmail(1M)` accepts. However, most of these flags have no effect, except for the `-C` option and three of the `Processing Options` that can be set from a configuration file with the `-o` option:

`-C` */path/to/alt/config/file*	Use alternate configuration file.
`-oA`*file*	Specify possible alias files.
`-oL`*n*	Set the default log level to *n*. Defaults to `9`.
`-on`	Validate the RHS of aliases when rebuilding the `aliases(4)` database.

`newaliases` runs in verbose mode (`-v` option) automatically.

使用例

例 1　Running the newaliases Command

The following command runs `newaliases` on an alias file different from the `/etc/mail/aliases` default in `sendmail(1M)`:

```
example% newaliases -oA/path/to/alternate/alias/file
```

終了ステータス

`newaliases` returns an exit status describing what it did. The codes are defined in `/usr/include/sysexits.h`.

`EX_OK`	Successful completion on all addresses.
`EX_NOUSER`	User name not recognized.
`EX_UNAVAILABLE`	Catchall. Necessary resources were not available.
`EX_SYNTAX`	Syntax error in address.
`EX_SOFTWARE`	Internal software error, including bad arguments.
`EX_OSERR`	Temporary operating system error, such as “cannot fork”.
`EX_NOHOST`	Host name not recognized.
`EX_TEMPFAIL`	Message could not be sent immediately, but was queued.

ファイル

`/etc/aliases`	Symbolic link to `/etc/mail/aliases`
`/etc/mail/aliases.pag`	
`/etc/mail/aliases.dir`	`ndbm` files maintained by `newaliases`
`/etc/mail/aliases.db`	Berkeley DataBase file maintained by newaliases

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

関連項目

sendmail(1M), aliases(4), attributes(5)

名前

newfs – construct a UFS file system

形式

`newfs [-NSBTv] [mkfs-options] raw-device`

機能説明

`newfs` is a friendly front-end to the `mkfs(1M)` program for making UFS file systems on disk partitions. `newfs` calculates the appropriate parameters to use and calls `mkfs`.

If run interactively (that is, standard input is a tty), `newfs` prompts for confirmation before making the file system.

If the `-N` option is not specified and the inodes of the device are not randomized, `newfs` calls `fsirand(1M)`.

You must be super-user or have appropriate write privileges to use this command.

Creating a Multiterabyte UFS File System

Keep the following limitations in mind when creating a multiterabyte UFS file system:

- *nbpi* is set to 1 Mbyte unless you specifically set it higher. You cannot set *nbpi* lower than 1 Mbyte on a multiterabyte UFS file system.
- *fragsize* is set equal to *bsize*.

オプション

The following options are supported:

`-N`
Print out the file system parameters that would be used to create the file system without actually creating the file system. `fsirand(1M)` is not called here.

`-S`
Sends to stdout a human-readable version of the superblock that would be used to create a filesystem with the specified configuration parameters.

`-B`
Sends to stdout a binary (machine-readable) version of the superblock that would be used to create a filesystem with the specified configuration parameters.

`-T`
Set the parameters of the file system to allow eventual growth to over a terabyte in total file system size. This option sets *fragsize* to be the same as *bsize*, and sets *nbpi* to 1 Mbyte, unless the `-i` option is used to make it even larger. If you use the `-f` or `-i` options to specify a *fragsize* or *nbpi* that is incompatible with this option, the user-supplied value of *fragsize* or *nbpi* is ignored.

`-v`
Verbose. `newfs` prints out its actions, including the parameters passed to `mkfs`.

mkfs-options
Options that override the default parameters are:

`-a` *apc*
The number of alternate sectors per cylinder to reserve for bad block replacement for SCSI devices only. The default is `0`.

This option is not applicable for disks with EFI labels and is ignored.

-b *bsize*
The logical block size of the file system in bytes, either 4096 or 8192. The default is 8192. The sun4u architecture does not support the 4096 block size.

-c *cgsize*
The number of cylinders per cylinder group, ranging from 16 to 256. The default is calculated by dividing the number of sectors in the file system by the number of sectors in a gigabyte. Then, the result is multiplied by 32. The default value is always between 16 and 256.

mkfs can override this value. See mkfs_ufs(1M) for details.

This option is not applicable for disks with EFI labels and is ignored.

-C *maxcontig*
The maximum number of logical blocks, belonging to one file, that are allocated contiguously. The default is calculated as follows:

maxcontig = *disk drive maximum transfer size / disk block size*

If the disk drive's maximum transfer size cannot be determined, the default value for maxcontig is calculated from kernel parameters as follows:

If maxphys is less than ufs_maxmaxphys, which is typically 1 Mbyte, then maxcontig is set to maxphys. Otherwise, maxcontig is set to ufs_maxmaxphys.

You can set maxcontig to any positive integer value.

The actual value will be the lesser of what has been specified and what the hardware supports.

You can subsequently change this parameter by using tunefs(1M).

-d *gap*
Rotational delay. This option is obsolete in the Solaris 10 release. The value is always set to 0, regardless of the input value.

-f *fragsize*
The smallest amount of disk space in bytes that can be allocated to a file. *fragsize* must be a power of 2 divisor of *bsize*, where:

bsize / *fragsize* is 1, 2, 4, or 8.

This means that if the logical block size is 4096, legal values for *fragsize* are 512, 1024, 2048, and 4096. When the logical block size is 8192, legal values are 1024, 2048, 4096, and 8192. The default value is 1024.

For file systems greater than 1 terabyte or for file systems created with the -T option, *fragsize* is forced to match block size (*bsize*).

-i *nbpi*

The number of bytes per inode, which specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the number of inodes to create.

This value should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used. To create more inodes, a smaller number should be given. The default for *nbpi* is as follows:

```
Disk size                Density

Less than 1GB            2048
Less than 2GB            4096
Less than 3GB            6144
3GB to 1 Tbyte           8192
Greater than 1 Tbyte
   or created with -T    1048576
```

The number of inodes can increase if the file system is expanded with the `growfs` command.

-m *free*

The minimum percentage of free space to maintain in the file system, between 0% and 99%, inclusively. This space is off-limits to users. Once the file system is filled to this threshold, only the super-user can continue writing to the file system.

The default is ((64 Mbytes/partition size) * 100), rounded down to the nearest integer and limited between 1% and 10%, inclusively.

This parameter can be subsequently changed using the `tunefs(1M)` command.

-n *nrpos*

The number of different rotational positions in which to divide a cylinder group. The default is `8`.

This option is not applicable for disks with EFI labels and is ignored.

-o *space* | *time*

The file system can either be instructed to try to minimize the *time* spent allocating blocks, or to try to minimize the *space* fragmentation on the disk. The default is *time*.

This parameter can subsequently be changed with the `tunefs(1M)` command.

-r *rpm*

The rotational speed of the disk in revolutions per minute. The default is driver- or device-specific.

Note that you specify *rpm* for `newfs` and *rps* for `mkfs`.

This option is not applicable for disks with EFI labels and is ignored.

`-s` *size*
The size of the file system in sectors. The default is to use the entire partition.

`-t` *ntrack*
The number of tracks per cylinder on the disk. The default is taken from the disk label.

This option is not applicable for disks with EFI labels and is ignored.

オペランド

The following operands are supported:

raw-device
The name of a raw special device residing in the `/dev` directory (for example, `/dev/rdsk/c0t0d0s6`) on which to create the file system.

使用法

See `largefile(5)` for the description of the behavior of `newfs` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

使用例

例 1 Displaying the Parameters for the Raw Special Device

The following example verbosely displays the parameters for the raw special device, `c0t0d0s6`. It does not actually create a new file system:

```
example# newfs -Nv /dev/rdsk/c0t0d0s6
mkfs -F ufs -o N /dev/rdsk/c0t0d0s6 1112940 54 15 8192 1024 16 10 60
2048 t 0 —1 8 /dev/rdsk/c0t0d0s6: 1112940 sectors in
1374 cylinders of 15 tracks, 54 sectors 569.8MB in 86 cyl
groups (16 c/g, 6.64MB/g, 3072 i/g) super-block backups
(for fsck -b #) at:
32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224, . . .
```

例 2 Creating a UFS File System That Will Eventually Be Grown to a Multiterabyte UFS File System

The following example creates a UFS file system that will eventually be grown to a multiterabyte UFS file system.

This command creates a 800-Gbyte file system on the volume, `/dev/md/rdsk/d99`.

```
# newfs -T /dev/md/rdsk/d99
newfs: construct a new file system /dev/md/rdsk/d99: (y/n)? y
   /dev/md/rdsk/d99: 1677754368 sectors in 45512 cylinders of
   144 tracks, 256 sectors
   819216.0MB in 1821 cyl groups (25 c/g, 450.00MB/g, 448 i/g) . . .
```

Then, if you increase the volume size for this file system, you can use the `growfs` command to expand the file system. The file system is grown to 1.2 terabytes in this example:

```
# growfs -v /dev/md/rdsk/d99
/usr/lib/fs/ufs/mkfs -G /dev/md/rdsk/d99 2516631552 /dev/md/rdsk/d99:
   2516631552 sectors in 68268 cylinders of 144 tracks, 256  sectors
   1228824.0MB in 2731 cyl groups (25 c/g, 450.00MB/g, 448 i/g). . .
```

終了ステータス

The following exit values are returned:

`0`
The operation was successful.

`1, 10`
Usage error or internal error. A message is output to `STDERR` explaining the error.

Other exit values may be returned by `mkfs(1M)`, which is called by `newfs`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`fsck(1M)`, `fsck_ufs(1M)`, `fsirand(1M)`, `mkfs(1M)`, `mkfs_ufs(1M)`, `tunefs(1M)`, `attributes(5)`, `largefile(5)`, `ufs(7FS)`

診断

`newfs: No such file or directory`
The device specified does not exist, or a disk partition was not specified.

special`: cannot open`
You must write access to the device to use this command.

名前

newkey – create a new Diffie-Hellman key pair in the publickey database

形式

```
newkey -h hostname [-s nis | files | ldap]
newkey -u username [-s nis | files | ldap]
```

機能説明

newkey establishes new public keys for users and machines on the network. These keys are needed when using secure RPC or secure NFS service.

newkey prompts for a password for the given *username* or hostname and then creates a new public/secret Diffie-Hellman 192 bit key pair for the user or host. The secret key is encrypted with the given password. The key pair can be stored in the /etc/publickey file or the NIS publickey map.

newkey consults the publickey entry in the name service switch configuration file (see nsswitch.conf(4)) to determine which naming service is used to store the secure RPC keys. If the publickey entry specifies a unique name service, newkey will add the key in the specified name service. However, if there are multiple name services listed, newkey cannot decide which source to update and will display an error message. The user is required to specify the source explicitly with the -s option.

In the case of NIS, newkey should be run by the superuser on the master NIS server for that domain.

In the case of LDAP, newkey should be run by the superuser on a machine that also recognizes the directory manager's bind distinguished name (DN) and password to perform an LDAP update for the host.

オプション

-h *hostname* Create a new public/secret key pair for the privileged user at the given hostname. Prompts for a password for the given hostname.

-u *username* Create a new public/secret key pair for the given *username*. Prompts for a password for the given *username*.

-s nis
-s files
-s ldap Update the database in the specified source: nis (for NIS), files, or ldap (LDAP). Other sources may be available in the future.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

chkey(1), keylogin(1), nsswitch.conf(4), publickey(4), attributes(5)

名前

nfs4cbd – NFS Version 4 コールバックデーモン

形式

/usr/lib/nfs/nfs4cbd

機能説明

nfs4cbd デーモンは、NFS Version 4 プロトコルのコールバックプログラムの通信エンドポイントを管理します。nfs4cbd は、NFS Version 4 クライアント上で動作し、コールバックを送信可能な各トランスポート用のリスナーポートを作成します。

nfs4cbd デーモンは、NFS Version 4 クライアントの排他的使用のために提供されています。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/nfs

関連項目

svcs(1), mount_nfs(1M), svcadm(1M), attributes(5), smf(5)

注意事項

nfs4cbd サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

svc:/network/nfs/cbd

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

無効になっている場合は、application/auto_enable プロパティーが false に設定されていないかぎり、初回の NFSv4 マウント時に mount_nfs(1M) および automountd(1M) を使用することで有効になります。

Solaris の将来のリリースでは、このデーモンが存在しなくなる可能性があります。

名前

nfsd – NFS デーモン

形式

```
/usr/lib/nfs/nfsd [-a] [-c #_conn] [-l listen_backlog]
      [-p protocol] [-t device] [nservers]
```

機能説明

nfsd は、クライアントファイルシステムの要求を処理するデーモンです。{PRIV_SYS_NFS} および /var/run に書き込めるだけの十分な特権を備えたユーザーだけが、このデーモンを実行できます。

nfsd デーモンは、share(1M) および -a オプションを使用して自動的に呼び出されます。

バージョン 2 および 3 の場合、デフォルトでは、nfsd は TCP および UDP トランスポート経由で起動します。バージョン 4 の場合、デフォルトでは TCP 経由で起動します。これは、-p オプションを使用すると変更できます。

以前に呼び出された nfsd デーモンは、オプション指定の有無にかかわらず、別の nfsd コマンドを呼び出す前に停止する必要があります。

nfsd の起動パラメータを変更するには、sharectl(1M) コマンドを使用します。

SMF による管理

nfsd サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/network/nfs/server
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

nfsd が SIGTERM により強制終了された場合は、サービス管理機能によって再起動されることはありません。代わりに、SIGINT などのほかのシグナルを使って nfsd を再起動できます。

sharectl(1M) コマンドを使用すると、nfsd 関連のすべてのパラメータを管理できます。現在サポートされているパラメータを次に示します。

server_versmax=*num*
server_versmin=*num*
: NFS サーバーは、これらの変数で指定された範囲の NFS バージョンのみを使用します。有効な値 (バージョン) は、2、 3、4 です。これらのパラメータのいずれかまたは両方が未設定の場合、デフォルトの最小バージョンは 2、デフォルトの最大バージョンは 4 になります。

server_delegation=on|off
: デフォルトでは、この変数は on で、NFS サーバーはクライアントに委託を付与します。ユーザーは、この変数を off (大文字と小文字は区別される) に設定することにより、エクスポートされたすべてのファイルシステムで委託を無効にできます。この変数は、NFS Version 4 にのみ適用されます。

max_connections=*num*
: 並行する接続型接続の最大数を設定します。デフォルトは無制限で、-1 に設定することで得られます。これは nfsd の -c オプションと同等です。

listen_backlog=*num*
: 接続型トランスポートを介した NFS の接続キューの長さを設定します。デフォルト値は 32 で、これはキュー内に 32 個のエントリがあることを意味します。これは nfsd の -l オプションと同等です。

protocol=ALL
: 指定したプロトコルのみを介して nfsd を起動します。これは、nfsd の -p オプションと同等です。ALL は、nfsd コマンド行の -a と同等です。NFS SMF のパラメータ device とは相互に排他的です。NFS SMF のパラメータ device と protocol のいずれか一方は、設定しないようにする必要があります。両方を設定すると、nfs/server サービスは保守モードに入ります。UDP プロトコルの場合は、バージョン 2 とバージョン 3 のサービスのみが確立されます。NFS Version 4 は、UDP プロトコルに用にはサポートされていません。これは -p オプションと同等です。

device=*devname*
: NFS デーモンを、指定したデバイスによるトランスポート専用で起動します。これは、nfsd の -t オプションと同等です。NFS SMF のパラメータ protocol とは相互に排他的です。NFS SMF のパラメータ device と protocol のいずれか一方は、設定しないようにする必要があります。

servers=*num*
: 並行 NFS 要求の最大数。これは、nfsd コマンド行の最後の数値引数と同等です。デフォルトは 1024 です。これは、*nservers* オペランドと同等です。

後述の「使用例」を参照してください。

オプション

次のオプションがサポートされています。

-a
: UDP と TCP を含む、利用可能なすべての非接続型および接続型のトランスポートを介して NFS デーモンを起動します。これは、NFS の SMF で、sharectl(1M) コマンドを使用して protocol パラメータを ALL に設定することと同等です。

-c #_*conn*
: これにより、接続型トランスポートを介して NFS サーバーに許可される接続の最大数を設定します。デフォルトでは、この接続の数は無制限です。これは、NFS の SMF で sharectl(1M) コマンドを使用して設定する max_connections パラメータと同等です。

-l
: 接続型トランスポートを介した NFS TCP の接続キューの長さを設定します。デフォルト値は 32 エントリです。これは、NFS の SMF で sharectl(1M) コマンドを使用して設定する listen_backlog パラメータと同等です。

-p *protocol* 指定したプロトコルを介して NFS デーモンを起動します。これは、NFS の SMF で sharectl(1M) コマンドを使用して設定する protocol パラメータと同等です。

-t *device* NFS デーモンを、特定のデバイスのトランスポート用に起動します。これは、NFS の SMF で sharectl(1M) コマンドを使用して設定する device パラメータと同等です。

オペランド

次のオペランドがサポートされています。

nservers これは、サーバーが処理可能な並行する NFS 要求の最大数を設定します。この並行性を実現するため、カーネル内で必要に応じて *nservers* スレッドまで作成されます。*nservers* は、このサーバーに予想される負荷に基づいて設定します。通常は、 *nservers* に 16 を設定します。*nservers* を指定しない場合、並行する NFS 要求の最大数はデフォルトの 1 になります。これは、NFS の SMF で sharectl(1M) コマンドを使用して設定する servers パラメータと同等です。

使用法

NFS_PORTMON 変数が /etc/system で設定されている場合、クライアントは特権ポート (ポートは IPPORT_RESERVED 未満) を使用して NFS サービスを取得する必要があります。デフォルトでは、この変数はゼロと等価です。この変数は、「nfs」モジュールから「nfssrv」モジュールに移動されました。この変数を設定するには、/etc/system ファイルを編集して次のエントリを追加します。

```
set nfssrv:nfs_portmon = 1
```

使用例

例 1 委託を無効にする

以降の例で指定する nfsd のプロパティーについては、前述の「SMF による管理」を参照してください。

委託とは、サーバーがファイルの管理をクライアントに委託する NFSv4 の機能です。たとえば、サーバーは、読み取り委託または書き込み委託のいずれかをクライアントに付与する (または付与しない) ことができます。次のコマンドは、server_delegation プロパティーを off に設定してこれを実行します。

```
# sharectl set -p server_delegation=off nfs
```

例 2 委託の値を判別する

次のコマンドは、server_delegation プロパティーの現在の値を取得します。

```
# sharectl get -p server_delegation nfs
server_delegation=on
```

例3 並行要求の最大数を設定する

次のコマンドは、並行する NFS 要求の最大数を設定します。

```
# sharectl set -p servers=32 nfs
```

例4 接続キューの長さを設定する

次のコマンドは、接続型トランスポートを介した NFS の最大キュー長を設定します。

```
# sharectl set -p listen_backlog=48 nfs
```

終了ステータス

0 デーモンが正常に起動されました。

1 デーモンの起動に失敗しました。

ファイル

`.nfs`*XXX*	開かれているがリンクが解除されているファイルへのクライアントマシンのポインタ。
`/etc/system`	システム構成情報ファイル。
`/var/nfs/v4_state` `/var/nfs/v4_oldstate`	サーバーがクライアント状態情報の管理に使用されるディレクトリ。これらのディレクトリは削除しないでください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/file-system/nfs

関連項目

ps(1), svcs(1), mountd(1M), share(1M), sharectl(1M), svcadm(1M), sharetab(4), system(4), attributes(5), smf(5)

『Oracle Solaris のシステム管理 (ネットワークサービス)』

注意事項

nfsd を手動で起動または再起動することは推奨されていません。手動で起動または再起動する必要がある場合は、svcadm を使用して nfs サービス (svc:/network/nfs/server) を有効または無効にしてください。無効になっている場合は、application/auto_enable プロパティーが false に設定されていないかぎり、share_nfs(1M) により有効になります。詳細については、『『Oracle Solaris のシステム管理 (ネットワークサービス)』』および svcadm(1M) を参照してください。

名前 nfslogd – nfs ロギングデーモン

形式 /usr/lib/nfs/nfslogd

機能説明 nfslogd デーモンは、Solaris NFS サーバーに操作可能なロギング機能を提供します。nfslogd デーモンの役割は、NFS サーバーにより処理された RPC 操作を分析して、アクティビティーログを生成することです。このログは、ロギングを有効にしてエクスポートされたファイルシステムでのみ生成されます。これは、ファイルシステムのエクスポート時に、share_nfs(1M) コマンドを使って指定されます。

NFS Version 4 を使用する Solaris マシンでは、NFS サーバーのロギングはサポートされていません。

ログファイル内の各レコードには、タイムスタンプ、クライアントシステムの IP アドレス (またはホスト名 (解決可能な場合))、操作対象のファイル名またはディレクトリ名、および操作のタイプが含まれています。基本書式では、操作は入力 (i) または出力 (o) 操作のいずれかになります。NFS サーバーログの基本書式は、Washington University FTPd デーモンにより生成されるログ書式と互換性があります。ログ書式を拡張して、mkdir、 rmdir、remove などのディレクトリ変更操作を含めることもできます。拡張された書式は、Washington University FTPd デーモンの書式とは互換性がありません。詳細については、nfslog.conf(4) を参照してください。

NFS サーバーのロギングメカニズムは、2 つのフェーズに分けられます。最初のフェーズは NFS カーネルモジュールにより実行され、raw RPC 要求およびその結果を持続ストレージによりバックアップされる作業バッファーに記録します。作業バッファーの場所は、/etc/nfs/nfslog.conf ファイルに指定されます。詳細は、nfslog.conf(4) を参照してください。2 番目のフェーズには、nfslogd ユーザーレベルデーモンが関係します。これは、作業バッファーを定期的に読み取り、raw RPC 情報を解釈し、関連する RPC 操作を単一のトランザクションレコードにまとめて、出力ログを生成します。その後、nfslogd デーモンは休眠状態に入り、作業バッファーにさらに情報が記録されるのを待機します。デーモンの休眠状態の長さは、/etc/default/nfslogd の IDLE_TIME パラメータを変更することで構成できます。作業バッファーは、nfslogd デーモン内部で使用するためのものです。

NFS 操作では、パス名の代わりにファイルハンドルが引数として使用されます。このため、nfslogd デーモンは、操作に関連するパス名を記録するために、対応するファイルハンドルではなく、ファイルハンドルからパスへのマッピングのデータベースを管理する必要があります。クライアントが検索や、ファイルハンドルをクライアントに返すほかの NFS 操作を実行すると、ファイルハンドルのエントリがデータベースに追加されます。

NFS クライアントがサーバーから取得したファイルハンドルは無期限に保持でき、あとでファイルやディレクトリに対する NFS 操作の引数として使用できます。NFS クライアントは、サーバーのリブート後でさえもそのファイルハンドルを使用できます。データベースは、サーバーのリブート後も利用可能でなければならないため、持続ストレージにバックアップされます。データベースの場所

は、/etc/nfs/nfslog.conf ファイル内の *fhtable* パラメータで指定されます。このデータベースは、 nfslogd デーモンにより内部で使用されます。

ファイルハンドルのマッピングデータベースを扱いやすいサイズに保つために、nfslogd は定期的にデータベースのプルーニングを行います。削除されるのは、指定された期間以上アクセスのないファイルハンドルのエントリです。プルーニングプロセスの実行間隔は、/etc/default/nfslogd 内の PRUNE_TIMEOUT 構成可能パラメータを使って指定します。前回のプルーニング処理の実行以降、使用されていないファイルハンドルのレコードは削除されます。データベースのプルーニングは、PRUNE_TIMEOUT を INT_MAX と同じ値に設定することで無効にできます。

プルーニングが有効な場合、クライアントがファイルハンドルを PRUNE_TIMEOUT よりも長期間保持し、マッピングデータベース内の一致するレコードが削除されたあとで、ファイルハンドルに対して NFS 操作を実行する危険が常に存在します。このような場合、ファイルハンドルのパス名は解決されず、ログにはパス名の代わりにファイルハンドルが含まれています。

nfslogd デーモンの動作に影響を及ぼすさまざまな構成可能パラメータが存在します。これらのパラメータは /etc/default/nfslogd に記述されています。詳細は後述します。

UMASK	ログファイル、作業バッファーファイル、およびファイルハンドルのマッピングデータベースのファイルモードを設定します。
MIN_PROCESSING_SIZE	作業情報を処理してログファイルに書き込む前に、バッファーファイルが達している必要のある最小サイズをバイト単位で指定します。MIN_PROCESSING_SIZE の値は、1 と ulimit の間にする必要があります。
IDLE_TIME	デーモンがバッファーファイルにさらに情報が記録されるのを待機する休眠期間を秒単位で指定します。IDLE_TIME は、構成ファイルを再読み込みする頻度も決定します。IDLE_TIME の値は、1 と INT_MAX の間にする必要があります。
MAX_LOGS_PRESERVE	nfslogd は、ログを定期的に切り替えます。MAX_LOGS_PRESERVE には、保存するログファイルの最大数を指定します。MAX_LOGS_PRESERVE に達すると、新しいログファイルの作成時にもっとも古いファイルが上書きされます。これらのファイルは、filename.0 から始まる番号付きの拡張子を付けて保存されます。拡張子に付けられた番号が最も大きいファイルが、もっとも古いファイルです。この番号の

最大値は、MAX_LOGS_PRESERVE に設定された値です。MAX_LOGS_PRESERVE の値は、1 と INT_MAX の間にする必要があります。

CYCLE_FREQUENCY
: ログファイルを切り替える頻度を時間単位で指定します。 CYCLE_FREQUENCY を使って、ログファイルが大きくなり過ぎないようにできます。CYCLE_FREQUENCY の値は、1 と INT_MAX の間にする必要があります。

MAPPING_UPDATE_INTERVAL
: ファイルハンドルからパスへのマッピングテーブル内のレコードを更新する間隔を秒単位で指定します。レコードへのアクセスが行われるたびにそのレコードの atime を更新する代わりに、このパラメータに設定された期間が経過したときにのみ更新します。プルーニングルーチンは、レコードへのアクセス時間を使用して、データベースからレコードを削除するかどうかを決定します。このパラメータの値は、1 と INT_MAX の間にする必要があります。

PRUNE_TIMEOUT
: データベースレコードがタイムアウトする期間を時間単位で指定します。最後にレコードにアクセスがあってからの経過時間が PRUNE_TIMEOUT より長い場合、そのレコードをデータベースからプルーニングできます。PRUNE_TIMEOUT のデフォルト値は 168 時間 (7 日) です。 PRUNE_TIMEOUT の値は、1 と INT_MAX の間にする必要があります。

終了ステータス

次の終了値が返されます。

0 デーモンが正常に起動されました。

1 デーモンの起動に失敗しました。

ファイル

/etc/nfs/nfslogtab

/etc/nfs/nfslog.conf

/etc/default/nfslogd

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/file-system/nfs

関連項目

share_nfs(1M), nfslog.conf(4), attributes(5)

名前

nfsmapid – NFS ユーザー ID とグループ ID のマッピングデーモン

形式

/usr/lib/nfs/nfsmapid

機能説明

nfsmapid デーモンは、NFS Version 4 の owner と owner_group の ID 属性と、NFS バージョン 4 のクライアントとサーバーの両方で使用されるローカル UID および GID 番号とのマッピングを行います。

nfsmapid は、/etc/nsswitch.conf ファイル内の passwd および group エントリを使用して、マッピングの実行方法を指示します。

nfsmapid デーモンには、ユーザーからアクセスできる外部インタフェースはありません。ただし、次のいずれかの方法を使って nfsmapid の構成を管理できます。

- NFS の SMF で、sharectl(1M) コマンドを使用して nfsmapid_domain パラメータを指定します。
- _nfsv4idmapdomain DNS リソースレコードを指定します。

現在選択されている NFSv4 ドメインは、/var/run/nfs4_domain ファイル内で使用できます。

詳細は、『『Oracle Solaris のシステム管理 (ネットワークサービス)』『Oracle Solaris のシステム管理 (ネットワークサービス)』』を参照してください。

SMF による管理

nfsmapid サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/network/nfs/mapid
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

無効になっている場合は、application/auto_enable プロパティーが false に設定されていないかぎり、mount_nfs(1M)、share_nfs(1M)、および automountd(1M) を使用することで有効になります。

nfsmapid は、ユーザーの UID と GID をキャッシュします。あとでユーザーが、後述するユーティリティーのいずれかを使って UID や GID を変更した場合、nfsmapid キャッシュは無効になります。この時点で、属性を取得または設定する NFS 操作では、この無効な情報が交換されます。この状況を解決するには、次のように nfsmapid を再起動します。

```
# svcadm restart svc:/network/nfs/mapid:default
```

ドメイン名を指定する起動 SMF パラメータ (nfsmapid_domain) は、sharectl(1M) コマンドを使って操作できます。

nfsmapid_domain

NFS SMF パラメータ nfsmapid_domain の設定は、送信と受信の各属性文字列の作成と比較に nfsmapid で使用されるドメインよりも優先されます。また、この設定は、NFSv4 ドメインを設定するほかのすべてのメカニズムよりも優先されます。nfsmapid_domain の設定が存在しない場合、nfsmapid デーモンは次のように NFSv4 ドメインを決定します。

- 適切に構成された /etc/resolv.conf (resolv.conf(4) を参照) が存在する場合、nfsmapid は指定されたネームサーバーに対してドメインを照会します。
- 適切に構成された /etc/resolv.conf (resolv.conf(4) を参照) が存在するが、照会先のネームサーバーが適切なドメイン名レコードを保持していない場合、nfsmapid は BIND インタフェースを介してドメイン名の取得を試みます (resolver(3RESOLV) を参照)。
- /etc/resolv.conf が存在しない場合、nfsmapid はフォールバックして構成済みのドメイン名 (domainname(1M) を参照) を使用します。返されるドメイン名は、先頭のドメイン接尾辞が削除されています。たとえば、widgets.sales.acme.com の場合は sales.acme.com が返されます。
- /etc/resolv.conf が存在せず、構成済みのドメイン名 (または /etc/defaultdomain) も存在しない場合は、ホスト名に完全指定のドメイン名 (FQDN) が含まれていれば、nfsmapid はフォールバックしてホスト名からドメイン名を取得します。

前述の手順をすべて実行してもドメイン名を取得できない場合、nfsmapid は構成済みのドメインを持ちません。その結果、次のような動作になります。

- owner および owner_group 送信属性文字列は、リテラル ID としてエンコードされます。たとえば、UID 12345 は 12345 としてエンコードされます。
- nfsmapid は、受信属性文字列の domain 部分を無視し、ユーザーまたはグループのネームサービス検索のみを実行します。ローカルシステムのネームサービスデータベース内にユーザー/グループが存在する場合は、ドメインが構成されていない場合でも、適切な UID/GID がマップされます。

この動作は、NFSv4 のクライアントとサーバー間で同じ管理ユーザー/グループドメインが存在する (つまり、クライアントとサーバーの両方でユーザー/グループの UID/GID が同じである) ことを意味します 重複する ID 空間では、受信属性文字列が不正な ID にマップされる可能性があります。ただし、これは、受信文字列を nobody にマップすることとは機能的に異なり、より高い柔軟性を実現します。後述の「使用例」を参照してください。

UID と GID を変更できるユーティリティーを次に示します。

- usermod(1M)
- userdel(1M)
- groupmod(1M)
- groupdel(1M)

ファイル

/var/run/nfs4_domain　現在 NFSv4 により使用されているドメイン名が含まれています。

使用例

例 1　ドメイン名を設定する

次のコマンドは、sharectl を使用してドメイン名を設定します。

```
# sharectl set -p nfsmapid_domain=sun.com nfs
```

nfsmapid_domain プロパティーについては、後述の「注意事項」を参照してください。

例 2　ドメイン名を取得する

次のコマンドは、sharectl を使用して現在のドメイン名を取得します。

```
# sharectl get -p nfsmapid_domain nfs
nfsmapid_domain=sun.com
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/nfs

関連項目

svcs(1), automountd(1M), domainname(1M), groupdel(1M), groupmod(1M), mount_nfs(1M), svcadm(1M), share_nfs(1M), sharectl(1M), userdel(1M), usermod(1M), resolver(3RESOLV), nfs(4), resolv.conf(4), attributes(5), smf(5)

『Oracle Solaris のシステム管理 (ネットワークサービス)』

注意事項

Solaris の将来のリリースでは、nfsmapid デーモンが存在しなくなる可能性があります。

名前

nfsref – NFS リフェラルユーティリティー

形式

```
nfsref [-t svc_type] add path location [location ...]

nfsref [-t svc_type] remove path

nfsref [-t svc_type] lookup path
```

機能説明

nfsref コマンドは、NFS Version 4 リフェラルを管理します。これはサーバー側のポインタで、クライアントをファイルシステムの実際の場所にリダイレクトするのに使用されます。リフェラルは、再解析ポイントに基づいています (reparsed(1M) および libreparse(3LIB) を参照)。どのような形式のパス引数も、再解析ポイントのシンボリックリンクへのパスを参照します。現在のところ、このコマンドは 1 つの「サービスタイプ」 nfs-basic を実装しています。これがデフォルトです。

このコマンドの最初の書式は nfsref add で、指定された場所へのリフェラルポイントを作成します。再解析ポイントが存在しない場合は、1 つ作成されます。存在する場合は、NFS サービスデータが追加されるか、既存の NFS サービスデータと置き換えられます。それぞれの場所の書式は、*host* :/*path* です。パスにはスペース文字を含めることができますが、シェルにより正しく解釈されるようにエスケープする必要があります。

このコマンドの 2 番目の書式は nfsref delete で、NFS リフェラルを削除します。これは、NFS サービスデータを指定された再解析ポイントから削除し、ほかのタイプのサービスデータが存在しない場合には再解析ポイントを削除します。

このコマンドの 3 番目の書式は nfsref lookup で、指定された NFS リフェラルポイントの場所を表示します。

オプション

次のオプションがサポートされています。

-t *svc_type*
: サービスタイプを指定します。現在のところ、nfs-basic のみがサポートされています。

使用例

例 1　NFS リフェラルを追加する

次のコマンドは、サーバーパス /pool/home/bob に、リソース homeserver:/homepool/bob を指す NFS リフェラルを作成します。

```
# nfsref add /pool/home/bob homeserver:/homepool/bob
Created reparse point /pool/home/bob

# nfsref lookup /pool/home/bob
homeserver:/homepool/bob
```

例 2　NFS リフェラルを削除する

次のコマンドは、サーバーパス /pool/home/bob で NFS リフェラルを削除します。

```
# nfsref remove /pool/home/bob homeserver:/homepool/bob
Removed svc_type 'nfs-basic' from /pool/home/bob
```

終了ステータス

`0`
正常終了。

`>0`
エラーが発生しました。

ファイル

`/usr/lib/reparsed/*.so.1`
`reparsed` のサービス別プラグイン。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	service/file-system/nfs
インタフェースの安定性	コンソリデーション非公開

関連項目

`reparsed(1M)`, `libreparse(3LIB)`, `attributes(5)`

名前

nfsstat – NFS 統計

形式

```
nfsstat [-cnrsza] [-T u | d ] [-v version] [interval [count]]

nfsstat -m [pathname]...
```

機能説明

nfsstat は、カーネルへのインタフェースである NFS および RPC (遠隔手続き呼び出し) の統計情報を表示します。また、nfsstat を使ってこの情報を再初期化することもできます。オプションを指定しない場合、デフォルトは次のようになります。

```
nfsstat -csnra
```

デフォルトではすべての情報を表示し、再初期化は行いません。

オプション

-a
: NFS_ACL 情報を表示します。

-c
: クライアント情報を表示します。クライアント側の NFS、RPC、および NFS_ACL 情報だけが出力されます。-n、-r、および -a オプションとともに使用すると、クライアント側の NFS、 RPC、および NFS_ACL 情報だけを出力できます。

-m [*pathname*...]
: NFS マウントされた各ファイルシステムの統計を表示します。*pathname* が指定されていない場合、NFS マウントされたすべてのファイルシステムの統計を表示します。*pathname* が指定されている場合、*pathname* で示される NFS マウントされたファイルシステムの統計を表示します。

  これには、サーバーの名前とアドレス、マウントフラグ、現在の読み取りと書き込みのサイズ、再転送カウント、属性キャッシュのタイムアウト値、フェイルオーバー情報、および動的再転送に使用されるタイマーが含まれています。動的再転送タイマーが表示されるのは、動的再転送が使用されている場合だけです。デフォルトでは、NFS は TCP プロトコルを介してマウントし、NFS バージョン 3 は TCP または UDP を介してマウントします。動的再転送は使用しません。

  -m オプションを指定すると、nfsstat はこのオプションだけを使用します。その他のオプションを -m とともに指定した場合、-m フラグをほかのオプションとともに指定できないことを警告するエラーメッセージが表示されます。

-n
: NFS 情報を表示します。クライアント側とサーバー側の NFS 情報が両方出力されます。-c および -s オプションとともに指定すると、クライアントまたはサーバーの NFS 情報のみを出力できます。

-r
: RPC 情報を表示します。

`-s`
サーバー情報を表示します。

`-T u|d`
タイムスタンプを表示します。

時刻の内部表現の出力表現に `u` を指定します。`time(2)` を参照してください。 標準の日付フォーマットに `d` を指定します。`date(1)` を参照してください。

`-v` *version*
統計を出力する NFS のバージョンを指定します。オプションの *version* 引数 (2|3|4) を続けて指定すると、そのバージョンの統計が出力されます。デフォルトでは、すべてのバージョンの統計が出力されます。

`-z`
ゼロ (再初期化) 統計。このオプションはスーパーユーザーのみが使用するものであり、上記の任意のオプションとともに設定することにより、特定の統計セットを出力後にゼロに初期化できます。

オペランド

次のオペランドがサポートされています。

count
count 回のみ報告します。

interval
各 interval 秒ごとに報告します。

pathname
NFS マウントされたファイルシステム内の、統計を表示するファイルのパス名を指定します。

表示

サーバー RPC の表示には、次のフィールドが含まれています。

`badcalls`
RPC レイヤーにより拒否された呼び出しの総数 (`badlen` と `xdrcall` の合計、詳細は後述)。

`badlen`
最小サイズの RPC 呼び出しよりも短い RPC 呼び出しの数。

`calls`
受信した RPC 呼び出しの総数。

`dupchecks`
重複要求キャッシュ内を検索する RPC 呼び出しの数。

`dupreqs`
重複が見つかった RPC 呼び出しの数。

`nullrecv`
RPC 呼び出しを受信したと考えられるのに、利用できなかった回数。

`xdrcall`
: ヘッダーを XDR で復号化できなかった RPC 呼び出しの数。

サーバー NFS のディスプレイには、受信した NFS 呼び出しの数 (`calls`) と拒否された NFS 呼び出しの数 (`badcalls`)、および作成されたさまざまな呼び出しの数と割合 (パーセンテージ) が表示されます。

サーバー NFS_ACL のディスプレイには、作成されたさまざまな呼び出しの数と割合 (%) が表示されます。

クライアント RPC の表示には、次のフィールドが含まれています。

`calls`
: 作成された RPC 呼び出しの総数。

`badcalls`
: RPC レイヤーにより拒否された呼び出しの総数。

`badverfs`
: 応答内のベリファイアが不正であるために呼び出しが失敗した回数。

`badxids`
: 未処理の呼び出しに対応していない応答をサーバーから受信した回数。

`cantconn`
: サーバーへの接続確立に失敗したために、呼び出しが失敗した回数。

`cantsend`
: クライアントが非接続型トランスポートを介して RPC 要求の送信を試みたが、送信できなかった回数。

`interrupts`
: 呼び出しが完了する前に、シグナルにより呼び出しが中断された回数。

`newcreds`
: 認証情報を書き換えなければならなかった回数

`nomem`
: メモリーの割り当てに失敗したために、呼び出しが失敗した回数。

`retrans`
: サーバーからの応答を待機中にタイムアウトしたために、呼び出しを再転送しなければならなかった回数。非接続型トランスポートを介した RPC にのみ適用されます。

`timeouts`
: サーバーからの応答を待機中に呼び出しがタイムアウトした回数。

`timers`
: 計算されたタイムアウト値が、呼び出しに対して指定された最小のタイムアウト値以上であった回数。

クライアント NFS のディスプレイには、送信されて拒否された呼び出しの数、CLIENT ハンドルを受信した回数 (clgets)、CLIENT ハンドルキャッシュに未使用のエントリが含まれていなかった回数 (cltoomany)、およびさまざまな呼び出しの数とその割合 (%) が表示されます。

クライアント NFS_ACL のディスプレイには、作成されたさまざまな呼び出しの数と割合 (%) が表示されます。

-m オプションには、マウントオプションで設定されたマウントフラグおよびシステム内部のマウントフラグに関する情報、およびその他のマウント情報が含まれています。mount_nfs(1M) を参照してください。

次のマウントフラグが、マウントオプションで設定されます。

forcedirectio
: クライアントとサーバー間で直接転送される、クライアントでバッファリングされないデータ。

grpid
: BSD グループ ID の継承。mount_nfs(1M) の説明を参照してください。

hard
: ハードマウント。

intr
: ハードマウントで割り込みが許可されます。

llock
: ローカルロックを使用中です (ロックマネージャーなし)。これは非公開インタフェースであることに注意してください。

noac
: クライアントは属性をキャッシュしません。

nointr
: ハードマウントで割り込みが許可されません。

nocto
: クローズからオープンへの整合性がありません。

retrans
: NFS 再転送。

rpctimesync
: RPC 時間同期。

rsize
: 読み取りバッファーのサイズ (バイト単位)

sec
: sec には、次のいずれかの値を指定できます。

dh
des - スタイルの認証 (タイムスタンプの暗号化)。

krb5
kerberos v5 - スタイルの認証。

krb5i
kerberos v5 - スタイルの認証 (整合性あり)。

krb5p
kerberos v5 - スタイルの認証 (非公開)。

none
認証なし。

short
UNIX スタイルの認証 (省略形)。

sys
UNIX スタイルの認証 (UID、GID)。

soft
ソフトマウント。

timeo
初期 NFS タイムアウト (1/10 秒単位)。

wsize
書き込みバッファーのサイズ (バイト単位)

次のマウントフラグは、システム内部用です。

acl
サーバーは NFS_ACL をサポートします。

down
サーバーがダウンしています。

dynamic
動的転送サイズ調整。

link
サーバーはリンクをサポートします。

mirrormount
mirrormount メカニズムにより自動的にマウントされます。

printed
「Not responding」というメッセージを出力します。

readdironly
readdirplus ではなく readdir を使用します。

`referral`
リフェラルメカニズムにより自動的にマウントされます。

`symlink`
サーバーはシンボリックリンクをサポートします。

追加のマウント情報に関連したフラグを次に示します。

`proto`
プロトコル。

`vers`
NFS のバージョン。

`-m` オプションでも、属性キャッシュのタイムアウト値を指定できます。`-m` 出力の次のフィールドに、属性キャッシュのタイムアウト値が表示されます。

`acdirmax`
キャッシュされたディレクトリ属性を保持する最大時間 (秒)。

`acdirmin`
キャッシュされたディレクトリ属性を保持する最小時間 (秒)。

`acregmax`
キャッシュされたファイル属性を保持する最大時間 (秒)。

`acregmin`
キャッシュされたファイル属性を保持する最小時間 (秒)。

`-m` 出力の次のフィールドに、フェイルオーバー情報が表示されます。

`currserver`
NFS サービスを提供中のサーバー。詳細については、『Oracle Solaris の管理: IP サービス』を参照してください。

`failover`
新規サーバーが選択された回数。

`noresponse`
サーバーが応答に失敗した回数。

`remap`
ファイルが新規サーバーに対して再評価された回数。

`-m` 出力の次のフィールドに、動的再転送に関する情報が表示されます。これらの項目が表示されるのは、動的再転送を使用している場合だけです。

`cur`
現在のバックオフ再転送値 (ミリ秒単位)。

`dev`
見積もられた偏差 (ミリ秒単位)。

`srtt`
平滑化された往復時間値 (ミリ秒単位)。

終了ステータス

次の終了値が返されます。

`0`
正常終了。

`>0`
エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/nfs

関連項目

`mount_nfs(1M)`, `attributes(5)`

『Oracle Solaris 11 システムのインストール』

『Oracle Solaris の管理: IP サービス』

名前

nscadm – ネットワークストレージ制御ユーティリティー

形式

`nscadm freeze` *device*

`nscadm unfreeze` *device*

`nscadm isfrozen` *device*

機能説明

`nscadm` コマンドは、いくつかのネットワークストレージ制御機能を実行します。

`nscadm freeze` コマンドは、指定されたデバイスへの既存の参照を閉じて、以降のアクセスを遮断します。これにより、システムをシャットダウンせずに、仮想ボリュームデバイスドライバ (RAID 0、RAID 1、RAID 5 など) を管理することが可能になります。

`nscadm unfreeze` コマンドは、指定したデバイスに対して `nscadm freeze` と逆の影響を及ぼします。

`nscadm isfrozen` コマンドは、指定されたデバイスの現在のステータスを返します。

オプション

`nscadm` コマンドでは次のオプションをサポートしています。

`-h` 使用方法のメニューを表示します。

オペランド

`nscadm` コマンド行では次のオプションをサポートしています。

device
`nscadm` の操作対象のストレージデバイスを指定します。

終了ステータス

`nscadm` とともに `freeze` や `unfreeze` サブコマンドを使用すると、次の終了値が返されます。

`0` 成功

`255` エラー

`nscadm` とともに `isfrozen` サブコマンドを使用すると、次の終了値が返されます。

`0` デバイスは現在凍結されています。

`1` デバイスは現在凍結されていません。

`255` エラー

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目 scmadm(1M), attributes(5)

名前

nscd – ネームサービスキャッシュデーモン

形式

```
/usr/sbin/nscd [-f configuration-file] [-g] [-e cachename, yes
 | no] [-i cachename]
```

機能説明

nscd デーモンは、大半のネームサービス要求用のキャッシュを提供するプロセスです。svc:/system/name-service/cache SMF サービスのサービスプロパティーは、キャッシュデーモンの動作を決定します。nscd.conf(4) を参照してください。

nscd provides caching for the passwd(4), group(4), hosts(4), ipnodes(4), exec_attr(4), prof_attr(4), user_attr(4), ethers(4), rpc(4), protocols(4), networks(4), bootparams(4), auth_attr(4), services(4), netmasks(4), project(4) databases through standard libc interfaces, such as gethostbyname(3NSL), getipnodebyname(3SOCKET), gethostbyaddr(3NSL), and others. ただし、シャドウファイルはキャッシュされません。結果として、getspnam(3C) の呼び出しはキャッシュされません。

データの生存期間は、キャッシュによって異なります。デフォルトでは、ローカルデータベース (/etc/hosts、/etc/passwd など) を変更すると、次回の nscd の呼び出し時にキャッシュが無効になります。

svc:/system/name-service/cache がオプションで依存する任意のサービス (以下に一覧表示) を更新および再表示すると、nscd が再起動します。これにより、すべてのキャッシュが効率的にクリアされます。

- svc:/network/dns/client (resolv.conf(4) を参照)
- svc:/network/nis/client (ypbind(1M) および ypfiles(4) を参照)
- svc:/network/ldap/client (ldapclient(1M) を参照)
- svc:/system/name-service/switch (nsswitch.conf(4) を参照)

nscd.conf(4) を参照してください。

nscd は、独自の管理ツールとしても動作します。nscd のインスタンスが稼働している場合は、コマンドが稼働中のバージョンに透過的に渡されます。

nscd は、ユーザーごとの検索が有効な状態 (nscd.conf(4) を参照) で稼働している場合に、要求を行うユーザーのために子プロセスを 1 つだけ生成します (ユーザーごとの nscd)。ユーザーごとの nscd は、ユーザーの資格情報を使って、ユーザーごとの検索スタイルに合わせて構成されたネームリポジトリへのユーザーごとの接続を開きます。検索は子プロセスで行われます。結果はプロセス内にキャッシュされ、同じユーザーだけが使用できます。キャッシュの管理方法は、メインの nscd デーモンが自身のキャッシュを管理する方法とまったく同じです。ユーザーからの以降の要求は、ユーザーごとの nscd が終了するまでは、その nscd により処理されます。ユーザーごとの nscd では、構成可能な非アクティブ生存期間 (TTL) の値が使用されるため、非アクティブ TTL を過ぎると終了します。

メインの nscd で作成可能なユーザーごとの nscd の最大数を指定できます (nscd.conf(4) を参照)。作成数が最大数を超過すると、メインの nscd は、必要に応じ、LRU アルゴリズムを使ってあまりアクティブでない子 nscd を終了します。

メインの nscd デーモンは、子 nscd をすべて作成、監視、および管理します。また、ユーザーごとの検索の初回受信時に、ユーザー自身の nscd を作成します。nscd デーモンの起動時にユーザーごとの検索が有効である場合、このデーモンは条件をすべて満たしていることを確認してから、ユーザーごとの nscd の作成を準備します。デーモンが停止すると、制御下にあるユーザーごとの nscd がすべて終了します。

ユーザーごとの nscd は、メインの nscd と同じ構成を使用します。同じデフォルト構成ファイルか、-f コマンド行オプションで指定されたものを読み込んで使用します。構成がいったん読み込まれると、ユーザーごとの nscd は、構成が有効な限りそれを使用します。

オプション

後述するオプションのいくつかでは、*cachename* の指定が必須です。*cachename* に指定可能な値には、passwd、group、hosts、ipnodes、exec_attr、prof_attr、user_attr、ethers、rpc、protocols、networks、bootparams、auth_attr、services、printers、project があります。

-f *configuration-file*
: nscd は、指定されたファイルから構成データを読み取ります。このオプションは廃止されており、将来のリリースで削除される予定です。

-g
: 現在の構成および統計を標準出力に出力します。これは、root 以外のユーザーが実行可能な唯一のオプションです。

-e *cachename*, yes|no
: 指定されたキャッシュを有効または無効にします。

-i *cachename*
: 指定されたキャッシュを無効にします。

使用例

例 1　nscd デーモンを停止して再起動します。

```
example# svcadm disable system/name-service/cache

example# svcadm enable system/name-service/cache
```

ファイル

/etc/nscd.conf　　廃止。以前はキャッシュデーモンの動作を決定していました。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

svcs(1), svcadm(1M), ypbind(1M), getspnam(3C), gethostbyname(3NSL), getipnodebyname(3SOCKET), auth_attr(4), bootparams(4), ethers(4), exec_attr(4),

group(4), hosts(4), netmasks(4), networks(4), nscd.conf(4), nsswitch.conf(4), passwd(4), prof_attr(4), project(4), protocols(4), resolv.conf(4), rpc(4), services(4), user_attr(4), attributes(5)

注意事項

nscd に -g オプションを指定した場合の出力は、変更される可能性があります。プログラミングインタフェースとして、これに依存しないようにしてください。

nscd サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/name-service/cache
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスステータスを照会するには、svcs(1) コマンドを使用します。

廃止されたサービス svc:/system/name-service-cache は、下位互換性を確保するために、参照される可能性のあるスクリプトとともに保持されています。 唯一の目的は、サービス svc:/system/name-service/cache への optional_all 依存として機能することです。廃止されたサービスの名前は、将来のリリースで削除されます。

名前

nscfg – import, export name service configurations

形式

```
/usr/sbin/nscfg command [cmd_options] [operands]
/usr/sbin/nscfg import [-fnvq] FMRI]
/usr/sbin/nscfg export [-nvq] FMRI]
/usr/sbin/nscfg unconfig [-nvq] FMRI]
/usr/sbin/nscfg validate [-vq] FMRI]
/usr/sbin/nscfg help
```

機能説明

The `nscfg` utility imports or exports legacy name service configuration files into or out of the SMF repository. Given a valid SMF configuration and corresponding FMRI, `nscfg` will regenerate the legacy naming service configuration files, such as `nsswitch.conf`, `resolv.conf`, `nscd.conf`, and so forth, into their legacy locations. Alternatively, `nscfg` can import those same configuration files populating the SMF repository if it is currently unpopulated.

The following are SMF services that can be operands to `nscfg`. In many instances, the legacy configuration file that is replaced by the SMF service is listed.

`svc:/system/name-service/switch:default`
: Legacy file: `/etc/nsswitch.conf`

  Name service switch configuration (used by `nscd`).

`svc:/system/name-service/cache:default`
: Legacy file: `/etc/nscd.conf`

  Name service cache (`nscd`).

`svc:/network/dns/client:default`
: Legacy file: `/etc/resolv.conf`

  DNS naming service.

`svc:/network/nis/domain:default`
: Legacy file: `/etc/defaultdomain`

  Legacy file: `/var/yp/binding/$DOMAIN/*`

  Shared NIS domain configuration. Used by all NIS services. Also (historical) shared use with LDAP naming services. Must be enabled when using `nis/client` or `ldap/client`.

`svc:/network/nis/client:default`
: NIS client naming service (`ypbind` and related)

`svc:/network/ldap/client:default`
: Legacy file: `/var/ldap/*`

  LDAP client naming service (`ldap_cachemgr` and related).

`svc:/network/nis/server:default`
NIS server service (`ypserv`).

`svc:/network/nis/passwd:default`
NIS server `passwd` service (`rpc.yppasswd`).

`svc:/network/nis/xfr:default`
NIS server `xfr` service (`ypxfrd`)

`svc:/network/nis/update:default`
NIS server update service (`rpc.ypupdated`)

`svc:/system/name-service/upgrade:default`
Import legacy configuration files into SMF service.

サブコマンド

The `nscfg` utility supports the subcommands described below. Options are described in the context of the subcommands.

`import` [`-fvq`] *FMRI*
If none of the SMF repository properties for the specified FMRI are currently populated, import the legacy configuration files associated with the specified FMRI into the SMF repository.

With `-f`, force the repopulation of the SMF repository with the legacy configuration, even if currently populated.

With `-v`, issue verbose progress messages during the requested operation. With `-q`, issue no error or other messages during the requested operation.

`export` [`-vq`] *FMRI*
Export the SMF configuration for the specified FMRI to legacy configuration files. This operation removes any existing affected legacy file(s) and generates new one(s) using the SMF configuration.

`unconfig` [`-vq`] *FMRI*
Unconfigure the SMF configuration for the specified FMRI. This operation resets the specified FMRI and any existing legacy file(s) to their initial unconfigured state.

With `-v`, issue verbose progress messages during the requested operation. With `-q`, issue no error or other messages during the requested operation.

`validate` [`-vq`] *FMRI*
Validate the SMF configuration for the specified FMRI. This operation checks the current configuration and verifies whether any errors exist.

With `-v`, issue verbose progress messages during the requested operation. With `-q`, issue no error or other messages during the requested operation.

使用例

例 1 Importing DNS Client Configuration

The following command imports the DNS client configuration, stored in `resolv.conf`, into the SMF repository.

```
# nscfg import svc:/network/dns/client:default
```

例 2 Exporting SMF LDAP Client Configuration

The following command exports the SMF LDAP client configuration to the legacy configuration files in `/var/ldap`.

```
# nscfg export svc:/network/ldap/client:default
```

例 3 Resetting Name Service Switch Configuration

The following command resets the name service switch configuration to its initial unconfigured state. The command generates no output and exits quietly with status only.

```
# nscfg unconfig -q svc:/system/name-service/switch:default
```

例 4 Validating LDAP Client Configuration

The following command validates the LDAP client configuration for errors or inconsistencies. The command generates no output and exits quietly with status only.

```
# nscfg validate -q svc:/network/ldap/client:default
```

終了ステータス

`0`
: Command successfully executed.

`1`
: An error occurred.

`2`
: Configuration unmodified, no change necessary.

`3`
: No configuration to import.

ファイル

- `/etc/default/{nss, yppasswd}`
- `/etc/defaultdomain`
- `/etc/nscd.conf`
- `/etc/nsswitch.conf`
- `/etc/resolv.conf`
- `/var/ldap/ldap_client_cred`
- `/var/ldap/ldap_client_file`
- `/var/yp/binding/{`*nisdomainname*`}/ypservers`
- `/var/yp/NISLDAPmapping`
- `/var/yp/securenets`
- `/var/yp/updaters`

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

関連項目

svcs(1), ldapclient(1M), nscd(1M), svcadm(1M), svccfg(1M), ypinit(1M), defaultdomain(4), nsswitch.conf(4), resolv.conf(4), ypfiles(4), attributes(5)

名前

nslookup – query Internet name servers interactively

形式

`nslookup [-option] [name | -] [server]`

機能説明

The `nslookup` utility is a program to query Internet domain name servers. It has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

パラメータ

Interactive mode is entered in the following cases:

1. No arguments are given (the default name server is used).
2. The first argument is a hyphen (-) and the second argument is the host name or Internet address of a name server.

Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

Options can also be specified on the command line if they precede the arguments and are prefixed with a hyphen. For example, to change the default query type to host information, and the initial timeout to 10 seconds, type:

```
nslookup -query=hinfo  -timeout=10
```

Interactive Commands

`host` [*server*]
: Look up information for host using the current default server or using server, if specified. If host is an Internet address and the query type is A or PTR, the name of the host is returned. If host is a name and does not have a trailing period, the search list is used to qualify the name. To look up a host not in the current domain, append a period to the name.

`server` *domain*
`lserver` *domain*
: Change the default server to *domain*; `lserver` uses the initial server to look up information about *domain*, while `server` uses the current default server. If an authoritative answer can't be found, the names of servers that might have the answer are returned.

`root`
: Not implemented.

`finger`
: Not implemented.

`ls`
: Not implemented.

`view`
: Not implemented.

`help`
Not implemented.

`?`
Not implemented.

`exit`
Exits the program.

`set` *keyword*[=*value*]
This command is used to change state information that affects the lookups. Valid keywords are:

`all`
Prints the current values of the frequently used options to `set`. Information about the current default server and host is also printed.

`class=`*value*
Change the query class to one of:

`IN`
the Internet class

`CH`
the Chaos class

`HS`
the Hesiod class

`ANY`
wildcard

The class specifies the protocol group of the information.

(Default = `IN`; abbreviation = `cl`)

`[no]debug`
Turn on or off the display of the full response packet and any intermediate response packets when searching.

(Default = `nodebug`; abbreviation = `[no]deb`)

`[no]d2`
Turn debugging mode on or off. This displays more about what `nslookup` is doing.

(Default = `nod2`)

`domain=`*name*
Sets the search list to *name*.

[no]search
: If the lookup request contains at least one period but doesn't end with a trailing period, append the domain names in the domain search list to the request until an answer is received.

(Default = search)

port=*value*
: Change the default TCP/UDP name server port to *value*.

(Default = 53; abbreviation = po)

querytype=*value*
type=*value*
: Change the top of the information query.

(Default = A; abbreviations = q, ty)

[no]recurse
: Tell the name server to query other servers if it does not have the information. (Default = recurse; abbreviation = [no]rec)

retry=*number*
: Set the number of retries to number.

timeout=*number*
: Change the initial timeout interval for waiting for a reply to number seconds.

[no]vc
: Always use a virtual circuit when sending requests to the server.

(Default = novc)

[no]fail
: Try the next nameserver if a nameserver responds with SERVFAIL or a referral (nofail) or terminate query (fail) on such a response.

(Default = nofail)

ファイル

/etc/resolv.conf
: resolver configuration file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/dns/bind
Interface Stability	Volatile

関連項目

dig(1M), host(1M), named(1M), attributes(5)

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at https://www.isc.org/software/bind/documentation.

注意事項

BIND 9 nslookup is deprecated and not as full featured as its BIND 8 version. For more features and functionality refer to dig(1M).

nslookup and dig(1M) now report “Not Implemented” as NOTIMP rather than NOTIMPL. This will have impact on scripts that are looking for NOTIMPL.

名前

nsupdate – 動的 DNS 更新ユーティリティー

形式

```
nsupdate [-dv] [-y keyname:secret | -k keyfile] [-t timeout]
      [-u udptimeout] [-r udpretries] [filename]
```

機能説明

nsupdate ユーティリティーは、RFC 2136 に定義されている動的 DNS 更新要求を、ネームサーバーに送信します。このユーティリティーを使用することで、ゾーンファイルを手動で編集しなくても、リソースレコードをゾーンに追加したりゾーンから削除したりできます。1 つの更新要求に、複数のリソースレコードの追加要求や削除要求を含めることができます。

nsupdate または DHCP サーバーにより動的に制御されているゾーンは、手動で編集しないようにしてください。手動で編集すると、動的更新との競合が発生して、データが失われる可能性があります。

nsupdate を使って動的に追加または削除されたリソースレコードは、同じゾーン内に存在する必要があります。要求は、ゾーンの SOA レコードの MNAME フィールドで識別されるゾーンのマスターサーバーに送信されます。

RFC 2845 に記述されている TSIG リソースレコードタイプを使用した動的 DNS 更新の認証には、トランザクション署名を利用できます。この署名が依存している共有シークレットを知るのは、nsupdate とネームサーバーだけです。現在のところ、TSIG でサポートされる暗号化アルゴリズムは、RFC 2104 で定義されている HMAC-MD5 だけです。その他のアルゴリズムが TSIG 用に定義される場合は、アプリケーションで相互の認証時に適切なアルゴリズムと鍵が選択されるようにする必要があります。たとえば、ネームサーバーが適切な秘密鍵とアルゴリズムを TSIG 認証を使用するクライアントアプリケーションの IP アドレスに関連付けることができるように、適切な key および server 文が /etc/named.conf に追加されます。nsupdate ユーティリティーは、/etc/named.conf を読み取りません。

nsupdate ユーティリティーとともに -y または -k オプションを使用して、TSIG レコードの生成に必要な共有シークレットを提供し、動的 DNS 更新要求を認証します。これらのオプションは相互に排他的です。「オプション」を参照してください。

オプション

次のオプションがサポートされています。

-d	デバッグモードで操作します。これにより、作成された更新要求およびネームサーバーから受信した応答に関するトレース情報が提供されます。
-k *keyfile*	*keyfile* ファイルから共有シークレットを読み取ります。名前の書式は次のとおりです。K{*name*}.+157.+{*random*}.private。歴史的な理由で、ファイル K{*name*}.+157.+{ *random*}.key も存在する必要があります。
-r *udpretries*	UDP での再試行回数を設定します。デフォルトの再試行回数は 3 です。*udpretries* がゼロに設定されている場合、更新要求が 1 つだけ作成されます。

-t*timeout*	更新が中止されるまでのタイムアウト間隔 (秒) を設定します。デフォルトは 300 秒です。ゼロに設定すると、タイムアウトが無効になります。
-u *udptimeout*	UDP での再試行までの間隔 (秒) を設定します。デフォルトは 3 秒です。ゼロに設定すると、間隔がタイムアウト (-t) と UDP での再試行回数 (-r) に基づいて計算されます。
-v	TCP 接続を使用します。更新要求のバッチを作成する場合は、TCP 接続の使用をお勧めします。デフォルトでは、nsupdate は UDP を使用して更新要求をネームサーバーに送信します。
-y *keyname*:*secret*	*keyname*:*secret* から署名を生成します。ここで、*keyname* は鍵の名前、*secret* は Base 64 でエンコードされた共有シークレットです。 -y オプションの使用は推奨されていません。これは、共有シークレットが平文のコマンド行引数として指定されるため、ps(1) の出力や、ユーザーのシェルで管理される履歴ファイルへの表示が可能であるためです。

入力形式

nsupdate ユーティリティーは、*filename* や標準入力から入力を読み取ります。各コマンドは、入力内に 1 行で指定されます。一部のコマンドは管理用です。その他のコマンドは、更新指示であるか、ゾーンの内容の前提条件チェックです。これらのチェックにより、名前またはリソースレコードのセット (RRset) がゾーンに存在するかどうかの条件が設定されます。更新要求全体を成功させるには、これらの条件を満たしている必要があります。前提条件のテストに失敗すると、更新は拒否されます。

すべての更新要求は、ゼロ以上の前提条件およびゼロ以上の更新で構成されます。この条件のために、指定されたリソースレコードがゾーンに存在または欠落している場合に、適切に認証された更新要求を処理することが可能になります。空白の入力行 (または send コマンド) が存在すると、累積されたコマンドが 1 つの動的 DNS 更新要求としてネームサーバーに送信されます。

このコマンドの書式とその意味は次のとおりです。

server *servername* [*port*]
: すべての動的更新要求をネームサーバー *servername* に送信します。server 文が指定されていない場合、nsupdate は更新を適切なゾーンのマスターサーバーに送信します。ゾーンの SOA レコードの MNAME フィールドにより、そのゾーンのマスターサーバーが特定されます。*port* 引数は、動的更新要求が送信される *servername* のポート番号です。ポート番号が指定されていない場合は、デフォルト DNS ポート番号 53 が使用されます。

`local` *address* [*port*]
ローカルアドレスを使用して、すべての動的更新要求を送信します。`local` 文が指定されていない場合、`nsupdate` はシステムが選択したアドレスとポートを使って更新を送信します。*port* 引数を使って、特定のポートからの要求を作成することもできます。ポート番号が指定されていない場合は、システムによりポート番号が割り当てられます。

`zone` *zonename*
すべての更新の対象をゾーン *zonename* に指定します。`zone` 文が指定されていない場合、`nsupdate` は残りの入力に基づいて適切な更新対象ゾーンの特定を試みます。

`class` *classname*
デフォルトクラスを指定します。クラスが指定されていない場合、デフォルトクラスは IN になります。

`key` *name secret*
すべての更新を *name secret* ペアを使用して TSIG で署名します。`key` コマンドは、`-y` や `-k` を使ってコマンド行で指定されたどの鍵よりも優先されます。

`prereq nxdomain` *domain-name*
domain-name という名前のリソースレコードが、どのタイプにも存在しないことを要求します。

`prereq yxdomain` *domain-name*
domain-name が存在することを要求します。domain-name には、いずれかのタイプのリソースレコードが 1 つ以上含まれている必要があります。

`prereq nxrrset` *domain-name* [*class*] *type*
指定された *type*、*class*、および *domain-name* のリソースレコードが存在しないことを要求します。*class* を省略すると、IN (インターネット) とみなされます。

`prereq yxrrset` *domain-name* [*class*] *type*
指定された *type*、*class*、および *domain-name* のリソースレコードが存在することを要求します。*class* を省略すると、IN (インターネット) とみなされます。

`prereq yxrrset` *domain-name* [*class*] *type data...*
共通の *type*、*class*、および *domain-name* を共有する、この書式の前提条件の各セットから取得された *data* は、結合されて RR セットになります。この RR セットは、ゾーン内に存在する *type*、*class*、*domain-name* で設定された既存の RR セットと正確に一致している必要があります。*data* は、リソースレコードの RDATA の標準テキスト表現で書き込まれます。

`update delete` *domain-name* [*ttl*] [*class*] [*type* [*data...*]]
domain-name という名前のリソースレコードをすべて削除します。*type* および *data* が指定されている場合は、一致するリソースレコードだけが削除されます。*class* が指定されていない場合は、インターネットクラスとみなされます。*ttl* は無視されます。これは、互換性を維持するためにのみ用意されています。

`update add` *domain-name ttl* [*class*] *type data...*
指定された *ttl*、*class*、および *data* の新規リソースレコードを追加します。

`show`
現在のメッセージを表示します。前回の送信時以降に指定されたすべての前提条件と更新が含まれています。

`send`
現在のメッセージを送信します。これは、空白行を挿入することと同等です。

`answer`
答えを表示します。

先頭がセミコロンの行はコメントであり、無視されます。

使用例

例 1 ゾーンに対してリソースレコードの挿入や削除を行う

この例では、`nsupdate` を使用して、`example.com` ゾーンに対してリソースレコードの挿入と削除を行う方法を示します。各例では、入力の末尾に空行が含まれることに注目してください。これは、コマンドのグループを 1 つの動的更新要求として `example.com` のマスターネームサーバーに送信するためです。

```
# nsupdate
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
> send
```

`oldhost.example.com` のすべての A レコードが削除されます。IP アドレス 172.16.1.1 の `newhost.example.com` の A レコードが追加されます。新たに追加されたレコードの TTL は 1 日 (86400 秒) です。

例 2 レコードが存在しない場合にのみ CNAME を追加する

次のコマンドは、レコードが存在しない場合にのみ CNAME を追加します。

```
# nsupdate
> prereq nxdomain nickname.example.com
> update add nickname.example.com 86400 CNAME somehost.example.com
> send
```

ネームサーバーは、前提条件に従って、どのタイプのリソースレコードも `nickname.example.com` に存在しないことを確認します。リソースレコードが存在する場合、更新要求は失敗します。この名前が存在しない場合は、`CNAME` が追加されます。この動作により、`CNAME` の追加時に、従来の RFC 1034 の規則 (名前が `CNAME` として存在する場合は、どのレコードタイプであれ、その名前がほかに存在してはならない) と矛盾することはありません。(この規則は RFC 4035 で DNSSEC 向けに更新され、`CNAME` が `RSIG`、`DNSKEY`、および `NSEC` レコードを保持することが可能になりました。)

ファイル

`/etc/resolv.conf`	デフォルトネームサーバーの識別に使用。
`K{name}.+157.+{random}.key`	`dnssec-keygen(1M)` により作成された HMAC-MD5 鍵の Base 64 エンコーディング
`K{name}.+157.+{random}.private`	`dnssec-keygen(1M)` により作成された HMAC-MD5 鍵の Base 64 エンコーディング

使用上の留意点

TSIG 鍵は 2 つの別個のファイルに重複して格納されます。これは、`nsupdate` で暗号化操作に DST ライブラリを使用したためで、将来のリリースでは変更される可能性があります。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	service/network/dns/bind
インタフェースの安定性	流動的

関連項目

`named(1M)`, `dnssec-keygen(1M)`, `attributes(5)`

RFC 2136、RFC 3007、RFC 2104、RFC 2845、RFC 1034、RFC 2535、RFC 2931、RFC 4035

名前

ntfscat – NTFS ファイルとストリームの標準出力への表示

形式

`ntfscat [options] device [file]`

機能説明

`ntfscat` コマンドは、ファイルまたはストリームを NTFS ボリュームから読み取り、その内容を標準出力に表示します。

`ntfscat` に渡されるファイル名では、大文字と小文字は区別されません。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-a`、`--attribute` *type*

特定の属性タイプの内容を表示します。デフォルトでは、名前のない `$DATA` 属性が表示されます。この属性は、10 進数、16 進数、または名前で指定できます。

```
Hex     Decimal Name
0x10    16      "$STANDARD_INFORMATION"
0x20    32      "$ATTRIBUTE_LIST"
0x30    48      "$FILE_NAME"
0x40    64      "$OBJECT_ID"
0x50    80      "$SECURITY_DESCRIPTOR"
0x60    96      "$VOLUME_NAME"
0x70    112     "$VOLUME_INFORMATION"
0x80    128     "$DATA"
0x90    144     "$INDEX_ROOT"
0xA0    160     "$INDEX_ALLOCATION"
0xB0    176     "$BITMAP"
0xC0    192     "$REPARSE_POINT"
0xD0    208     "$EA_INFORMATION"
0xE0    224     "$EA"
0xF0    240     "$PROPERTY_SET"
0x100   256     "$LOGGED_UTILITY_STREAM"
```

属性名は、先頭のドル記号 (`$`) なしで指定できます。`$` 記号を使用する場合は、シェルにより名前が解釈されないように、名前を引用符で囲む必要があります。

`-f`、`--force`

マウントされたボリュームを使用しないなど、一部の妥当なデフォルト値を無効にします。このオプションは慎重に使用してください。

`-h`、`--help`

それぞれの簡単な説明が付いたオプションのリストを表示します。

`-i`、`--inode` *num*

名前の代わりに i ノード番号を使ってファイルを指定します。

`-n、 --attribute-name` *name*
name で識別される属性を表示します。

`-q、 --quiet`
一部のデバッグ、警告、およびエラーメッセージを抑制します。

`-V、 --version`
バージョン番号、著作権、およびライセンス情報を表示します。

`-v、 --verbose`
追加のデバッグ、警告、およびエラーメッセージを表示します。

使用例

例 1 ルートにあるファイルの内容を表示する

次のコマンドは、NTFS ボリュームのルートにあるファイルの内容を表示します。

```
# ntfscat /dev/dsk/c0d0p1 boot.ini
```

例 2 サブディレクトリにあるファイルの内容を表示する

次のコマンドは、NTFS ボリュームのサブディレクトリ内にあるファイルの内容を表示します。

```
# ntfscat /dev/dsk/c0d0p1 /winnt/system32/drivers/etc/hosts
```

例 3 属性の内容を表示する

次のコマンドは、ルートディレクトリ (i ノード 5) の `$INDEX_ROOT` 属性の内容を表示します。

```
# ntfscat /dev/dsk/c0d0p1 -a INDEX_ROOT -i 5
```

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

`ntfsls(1M)`, `ntfsprogs(1M)`, `parted(1M)`, `attributes(5)`

http://wiki.linux-ntfs.org

著者

`ntfscat` は、Richard Russon、 Anton Altaparmakov、および Szabolcs Szakacsits が作成しました。

名前

ntfsclone – NTFS のクローン作成、イメージ作成、復元、またはレスキュー

形式

```
ntfsclone [options] source
ntfsclone --save-image [options] source
ntfsclone --resotore-image [options] source
ntfsclone --metadata [options] source
```

機能説明

`ntfsclone` ユーティリティーは効率的に、疎ファイル、イメージ、デバイス (パーティション)、または標準出力に、NTFS ファイルシステムのクローンを作成したり (コピー、保存、バックアップ、復元操作を含む)、NTFS ファイルシステムをレスキューしたりします。ディスクセクターレベルで機能し、書き込まれたデータだけ (つまり、空の領域以外) をコピーします。未使用のディスク領域は、0 になるか (疎ファイルにクローンを作成する場合)、制御コードで符号化されるか (特殊イメージ形式で保存する場合)、変更されずに残されるか (ディスク/パーティションにクローンを作成する場合)、または 0 で埋められます (標準出力にクローンを作成する場合)。

`ntfsclone` は、バックアップを作成 (NTFS ファイルシステムの正確なスナップショットを取得) して、あとから復元する場合に役立ちます。また、NTFS 読み取り/書き込み機能をテストする場合にも使用できます。問題の解決にクローンを使用すれば、元のファイルシステムを破壊する心配がありません。

特殊イメージ形式 (後述の同名の節を参照) を使用しない場合、クローンは、元の NTFS ファイルシステムをセクターレベルで正確にコピーしたものになります。したがって、元の NTFS ファイルシステムとまったく同様にマウントすることもできます。たとえば、ファイルにクローンを作成し、カーネルがループバックデバイスおよび NTFS をサポートしている場合、そのファイルは、次のコマンドを使用してマウントできます。

```
# mount -t ntfs -o loop ntfsclone.img
```

Windows のクローン作成

システムまたは起動パーティションを、別のコンピュータまたは異なるディスクやパーティションに (たとえば `/dev/dsk/c0d0p1` から `/dev/dsk/c0d0p2` へ、`/dev/dsk/c0d0p1` から `/dev/dsk/c0d1p1` へ、または別のディスクセクターオフセットへ)、コピー、移動、または復元する場合は、十分に注意する必要があります。

ほとんどの環境では、Windows を起動するために、次の特性が元のパーティションおよびディスクと同じであるパーティションに、NTFS をコピー、移動、または復元する必要があります。

- 開始セクターが同じ
- ディスクのタイプが同じ
- BIOS レガシーシリンダ設定が同じ

ntfscloneユーティリティーは、NTFSの正確なコピーを保証しますが、起動の問題には対処しません。これは仕様です。ntfscloneはシステムユーティリティーではなくファイルシステムユーティリティーです。NTFSのクローン作成だけを目的としており、Windowsのクローン作成は目的としていません。このため、ntfscloneは、Windowsのクローン作成用の非常に高速で信頼性の高い基本要素として使用できますが、完璧な解決策にはなりません。http://wiki.linux-ntfs.orgのNTFS Webサイトで、NTFSのクローン作成に役立つヒントを見つけることができます。

疎ファイル

未割り当てブロック(穴)が含まれるファイルは「疎ファイル」と呼ばれます。このようなファイルは常に、報告されるサイズが、実際に消費しているディスク容量よりも大きくなります。du(1)コマンドは、疎ファイルが使用する実際のディスク容量を報告します。穴は常に0として読み取られます。ext2、ext3、reiserfs、Reiser4、JFS、XFSなどの主要なLinuxファイルシステムはすべて、疎ファイルをサポートしています。ただし、ISO 9600 CD-ROMファイルシステムなど、サポートしていないものもあります。

特殊イメージ形式

NTFSファイルシステムは、特殊イメージ形式で保存することをお勧めします。未割り当てブロックは穴として表されず、制御コードを使用して符号化されます。したがって、疎ファイルをサポートしていなくても、イメージに領域が保存されます。このイメージ形式は、ファイルシステムイメージをネットワークでストリーミングする場合に最適です。特殊イメージ形式のデメリットは、イメージを直接マウントできないという点です。最初に復元しておく必要があります。

特殊イメージ形式を使用してイメージを保存するには、-sまたは--save-imageオプションを使用します。イメージを復元するには、-rまたは--restore-imageオプションを使用します。ソースファイルとしてハイフン(-)を使用すると、標準入力からイメージを復元できます。

メタデータのみのクローン作成

-mまたは--metadataオプションを使用すると、ntfscloneはNTFSメタデータのみを保存することができ、クローンを引き続きマウントできます。この使用法では、メタデータ以外のファイル内容がすべて失われ、データを読み戻すと、すべてが0になります。

メタデータのみのイメージは高い圧縮率で圧縮できます。通常、1 - 8Mバイトの範囲のサイズに圧縮されます。このことは、このようなイメージを調査やトラブルシューティングのために転送する場合に役立ちます。

メタデータのみのモードでは、ntfscloneは、メタデータに埋め込まれた常駐ユーザーデータなど、一切のユーザーデータを保存しません。すべて0で埋められます。さらに、すべてのファイルタイムスタンプと、メタデータ内の削除済み領域および未使用領域が、0で埋められます。したがってこのモードは、フォレンジック分析などには適していません。

ファイル名は削除されません。ファイル名には機密情報が含まれる場合があるので、メタデータのみのイメージを送信する前に、セキュリティーまたはプライバシーの侵害の可能性を考慮してください。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-f`、`--force`
整合性検査によりファイルシステムに「ダーティー」のマークが付けられている場合でも、`ntfsclone` を強制的に続行させます。

`-h`、`--help`
それぞれの簡単な説明が付いたオプションのリストを表示します。

`-i`、`--ignore-fs-check`
ファイルシステムの整合性検査の結果を無視します。このオプションは、`--meta-data` オプションとのみ一緒に使用できます。不一致の原因となるすべてのクラスタが保存されます。

`-m`、`--metadata`
メタデータのみのクローンを作成します。このオプションを指定する場合、ファイルにのみクローンを作成する必要があります。

`-o`、`--output` *file*
存在しない *file* に NTFS のクローンを作成します。*file* がハイフン (`-`) の場合は、クローンは標準出力に作成されます。

`-O`、`--overwrite` *file*
NTFS のクローンを *file* に作成します。*file* がすでに存在する場合はこれを上書きします。

`--rescue`
ディスク読み取りエラーを無視します。不良セクターが含まれるディスク (障害の発生したディスクなど) を、ディスクへの影響を最小限に抑えてレスキューできるようになります。このモードでは、`ntfsclone` は、より多くのデータをレスキューできるように最低のセクターレベルで動作します。読み取り不可能なセクターの内容は疑問符 (`?`) 文字で埋められます。このようなセクターの先頭には、`BadSector` という文字列のマークが付けられます。

`-r`、`--restore-image` *source*
source で指定された特殊イメージ形式から復元します。*source* がハイフン (`-`) の場合は、イメージは標準入力から読み取られます。

`-s`、`--save-image`
特殊イメージ形式で保存します。イメージを標準出力に作成する場合、領域の使用率および速度の点で、このオプションがもっとも効率的です。このオプションは、イメージの圧縮、暗号化、またはネットワークでのストリーミングに役立ちます。

使用例

例 1　上書きオプションを使用してクローンを作成する

次のコマンドは、--overwrite オプションを使用してクローンを作成します。

```
# ntfsclone --overwrite /dev/dsk/c0d2p1 /dev/dsk/c0d0p1
```

例 2　特殊イメージ形式で保存する

次のコマンドは、元のパーティションに特殊イメージ形式でクローンを作成します。

```
# ntfsclone --save-image --output backup.img /dev/dsk/c0d0p1
```

例 3　特殊イメージファイルから復元する

次のコマンドは、特殊イメージファイルから NTFS を復元します。

```
# ntfsclone --restore-image --overwrite /dev/dsk/c0d0p1 backup.img
```

例 4　圧縮イメージに保存する

次のコマンドは、圧縮イメージファイルに NTFS を保存します。

```
# ntfsclone --save-image -o - /dev/dsk/c0d0p1
```

例 5　圧縮イメージから復元する

次のコマンドは、圧縮イメージファイルから NTFS ボリュームを復元します。

```
# gunzip -c backup.img.gz | \
ntfsclone --restore-image --overwrite /dev/dsk/c0d0p1 -
```

例 6　ssh を使用して遠隔ホストにバックアップを作成する

次のコマンドは、ssh(1) を使用して遠隔ホストにバックアップを作成します。ssh にはパスワードが必要な場合があります。

```
# ntfsclone --save-image --output - /dev/dsk/c0d0p1 | \
gzip -c | ssh host ‘cat > backup.img.gz‘
```

例 7　ssh を使用して遠隔ホストから復元する

次のコマンドは、ssh(1) を使用して遠隔ホストから復元します。ssh にはパスワードが必要な場合があります。

```
# ssh host ‘cat backup.img.gz‘ | gunzip -c  | \
ntfsclone --restore-image --overwrite /dev/dsk/c0d0p1 -
```

例 8　Web サーバーからイメージファイルをストリーミングする

次のコマンドは、Web サーバーからイメージファイルをストリーミングして、パーティションに復元します。

```
# wget -qO - http://server/backup.img | \
ntfsclone --restore-image --overwrite /dev/dsk/c0d0p1 -
```

例 9 新しいファイルにクローンを作成する

次のコマンドは、存在しないファイルに NTFS ボリュームのクローンを作成します。

```
# ntfsclone --output ntfs-clone.img /dev/dsk/c0d0p1
```

例 10 NTFS メタデータを圧縮する

次のコマンドは、NTFS メタデータをイメージファイルに圧縮します。bzip2 は gzip よりも時間がかかりますが、圧縮後のアーカイブのサイズは gzip に比べて最高で 10 分の 1 になります。

```
# ntfsclone --metadata --output ntfsmeta.img /dev/dsk/c0d0p1
bzip2 ntfsmeta.img
```

例 11 NTFS メタデータを展開する

次のコマンドは、NTFS メタデータを疎ファイルに展開します。

```
# bunzip2 -c ntfsmeta.img.bz2 | \
cp --sparse=always /proc/self/fd/0 ntfsmeta.img
```

終了ステータス

リターンコードは成功した場合は 0 に、失敗した場合は 0 以外になります。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

du(1), ssh(1), ntfsresize(1M), ntfsundelete(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfsclone は、Per Olofsson (特殊イメージ形式のサポート) と Anton Altaparmakov の協力の下、Szabolcs Szakacsits が作成しました。

名前

ntfscluster – NTFS ボリュームの指定領域におけるファイルの識別

形式

`ntfscluster [options] device`

機能説明

`ntfscluster` ユーティリティーには、次に説明する `info`、`sector`、および `cluster` の 3 つのモードがあります。

`info`
: デフォルトモードである `info` は、現在実装されていません。このモードが有効になっている場合、NTFS ボリュームに関する一般情報が表示されます。

`sector`
: `sector` モードでは、指定した範囲のセクターにデータが存在するファイルのリストが表示されます。このモードは、`--sector` オプションで有効になります。

`cluster`
: `cluster` モードでは、指定した範囲のクラスタにデータが存在するファイルのリストが表示されます。クラスタサイズが 1 セクターである場合は、`sector` 操作モードと同等です。このモードは、`--cluster` オプションで有効になります。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-c`、`--cluster` *range*
: この範囲のクラスタにデータが存在するすべてのファイルが表示されます。

`-F`、`--filename` *filename*
: *filename* に関する情報を表示します。

`-f`、`--force`
: マウントされたボリュームを扱わないなど、一部の妥当なデフォルト値をオーバーライドします。このオプションは慎重に使用してください。

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-I`、`--inode` *num*
: この i ノードに関する情報を表示します。

`-i`、`--info`
: このオプションはまだ実装されていません。

`-q`、`--quiet`
: 一部のデバッグ、警告、およびエラーメッセージを抑制します。

`-s`、`--sector` *range*
: この範囲のセクターにデータが存在するすべてのファイルが表示されます。

`-V`、`--version`
: バージョン番号、著作権、およびライセンス情報を表示します。

-v、--verbose
追加のデバッグ、警告、およびエラーメッセージを表示します。

使用例

例1 ボリュームに関する情報を表示する

次のコマンドは、ボリューム /dev/dsk/c0d0p1 に関する情報を表示します。

```
# ntfscluster /dev/dsk/c0d0p1
```

例2 クラスタ範囲に存在するファイルのリストを表示する

次のコマンドは、/dev/dsk/c0d0p1 の最初の 500 のクラスタに存在するファイルを検索します。

```
# ntfscluster -c 0-500 /dev/dsk/c0d0p1
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

ntfsinfo(1M), ntfsprogs(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfscluster は、Anton Altaparmakov の協力の下、Richard Russon が作成しました。

名前

ntfscmp – 2 つの NTFS ファイルシステムの比較と相違点の報告

形式

`ntfscmp [options] device1 device2`

機能説明

`ntfscmp` ユーティリティーは、2 つの NTFS ファイルシステムのあらゆる側面を比較して、見つかったすべての相違点を報告します。ファイルシステムは、ブロックデバイスまたはイメージファイル上のものにできます。`ntfscmp` はボリューム検証に使用できます。ただし、その主な目的は、効率的な開発ツールとして、NTFS に行われたメタデータ変更を迅速に検出し、特定して、その妥当性を調べることです。

NTFS メタデータの変更にのみ関心がある場合は、`ntfsclone(1M)` の `--metadata` オプションを使用して作成したメタデータイメージを比較すると便利です。このようにすると、通常は確認する必要がないタイムスタンプの変更を除外できます。

`ntfscmp` の出力は簡潔ですが、相違点を正確に判断するための情報としては十分です。`diff(1)` を使用して、報告された i ノードごとに `ntfsinfo(1M)` の詳細な出力を比較すれば、より豊富な出力が得られます。

オプション

サポートされているオプションを次に示します。オプションには、単一文字と完全名の両方の形式があります。

`-h、--help`
: ヘルプを表示して終了します。

`-P、--no-progress-bar`
: 進捗状況を表示しません。

`-v、--verbose`
: 追加のデバッグ、警告、およびエラーメッセージを表示します。

終了ステータス

終了コードは成功した場合は `0` に、失敗した場合には 0 以外になります。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

`diff(1)`, `ntfscat(1M)`, `ntfsclone(1M)`, `ntfsinfo(1M)`, `ntfsprogs(1M)`, `parted(1M)`, `attributes(5)`

`http://wiki.linux-ntfs.org`

著者

`ntfscmp` は、Szabolcs Szakacsits が作成しました。

名前　ntfscp – NTFS ボリュームへのファイルのコピー

形式　`ntfscp [options] device source_file destination`

機能説明　`ntfscp` ユーティリティーは、ファイルを NTFS ボリュームにコピーします。*destination* (「形式」を参照) は、ファイルにすることもディレクトリにすることもできます。*destination* は名前で指定されたディレクトリであり、*source_file* はこのディレクトリにコピーされます。*destination* が i ノード番号で指定されたディレクトリである場合は、この i ノードに対して名前のないデータ属性が作成され、そこに *source_file* がコピーされます。i ノード番号でディレクトリを指定する前に、マイナスの結果になる可能性も考慮してください。通常、名前のないデータストリームはディレクトリに存在しません。

データストリーム　NTFS のすべてのデータはストリームに保存され、ストリームには名前を付けられます。1 つのファイルに複数のデータストリームを格納できますが、名前のないストリームが正確に 1 つ必要になります。ファイルのサイズは、その名前のないデータストリームのサイズになります。通常、ストリーム名を指定しない場合、名前のないデータストリームにアクセスすることになります。名前のあるデータストリームにアクセスする場合は、`:`*stream_name* をファイル名に追加する必要があります。たとえば、`some.mp3:artist` を開くと、`some.mp3` 内の `artist` ストリームが開かれます。名前が付けられているデータストリームへのアクセスを禁止しているオペレーティングシステム (Windows など) でこれらのストリームにアクセスするには、`FAR` などのプログラムや、`cygwin` のユーティリティーを使用する必要があります。

オプション　サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-a`、`--attribute` *num*
: *num* で指定された属性に書き込みます。

`-f`、`--force`
: マウントされたボリュームを扱わないなど、一部の妥当なデフォルト値を無効にします。このオプションは慎重に使用してください。

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-i`、`--inode`
: *destination* (「形式」を参照) を i ノード番号として扱います。

`-N`、`--attr-name` *name*
: この名前を持つ属性に書き込みます。

`-n`、`--no-action`
: 実際のコピー操作を行う前にテストを実行する場合に、このオプションを使用します。ボリュームは読み取り専用で開かれ、書き込みは行われません。

-q、 --quiet
一部のデバッグ、警告、およびエラーメッセージを抑制します。

-V、 --version
バージョン番号、著作権、およびライセンス情報を表示します。

-v、 --verbose
追加のデバッグ、警告、およびエラーメッセージを表示します。

使用例

例 1 ホームからルートディレクトリにコピーする

次のコマンドは、new_boot.ini を boot.ini として、/home/user から /dev/dsk/c0d0p1 の NTFS ボリュームのルートにコピーします。

```
# ntfscp /dev/dsk/c0d0p1 /home/user/new_boot.ini boot.ini
```

例 2 ストリームをコピーする

次のコマンドは、myfile を C:\some\path\myfile:stream (指定された /dev/dsk/c0d0p1 ドライブ指示子は C と想定) にコピーします。

```
# ntfscp -N stream /dev/dsk/c0d0p1 myfile /some/path
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

ntfsresize(1M), ntfsprogs(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfscp は、Anton Altaparmakov と Hil Liao の協力の下、Yura Pakhuchiy が作成しました。

名前

ntfsfix – 一般的なエラーの修復と、オペレーティングシステムでの NTFS の検査の強制

形式

`ntfsfix [options] device`

機能説明

`ntfsfix` ユーティリティーは、一般的な NTFS 問題のいくつかを修復します。`chkdsk` の一バージョンではありません。根本的な NTFS の不一致のいくつかを修復し、NTFS ジャーナルファイルをリセットし、オペレーティングシステムが次回リブートしたときに NTFS の整合性検査を行うようにスケジュールを設定します。

NTFS ボリュームがオペレーティングシステムまたは何らかのほかの原因によって損傷を受けた可能性があり、マウントできない場合は、そのボリュームで `ntfsfix` を実行します。

オプション

サポートされているオプションを次に示します。

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-V`、`--version`
: バージョン番号、著作権、およびライセンス情報を表示します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

`mkntfs(1M)`, `ntfsprogs(1M)`, `parted(1M)`, `attributes(5)`

http://wiki.linux-ntfs.org

著者

`ntfsfix` は、Szabolcs Szakacsits の協力の下、Anton Altaparmakov が作成しました。

名前

ntfsinfo – ファイルの属性のダンプ

形式

`ntfsinfo [options] device`

機能説明

`ntfsinfo` ユーティリティーは、i ノード *inode-number* またはファイル *path-filename* の属性や、MFT に関する情報 (`-m` オプション) をダンプします。オプションの全リストを表示するには、引数を付けずに、`ntfsinfo` を実行します。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-F`、`--file` *file*
: *file* に関する情報を表示します。

`-f`、`--force`
: 既存のファイルを上書きしないなど、一部の妥当なデフォルト値をオーバーライドします。このオプションは慎重に使用してください。

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-i`、`--inode` *num*
: *num* で識別された i ノードに関する情報を表示します。

`-m`、`--mft`
: ボリュームに関する情報を表示します。

`-q`、`--quiet`
: 一部のデバッグ、警告、およびエラーメッセージを抑制します。

`-t`、`--notime`
: 出力にタイムスタンプを表示しません。

`-V`、`--version`
: バージョン番号、著作権、およびライセンス情報を表示します。

`-v`、`--verbose`
: 追加のデバッグ、警告、およびエラーメッセージを表示します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

ntfsprogs(1M), ntfsundelete(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfsinfo は、 Matthew J. Fanto、 Anton Altaparmakov、 Richard Russon、 Szabolcs Szakacsits、 Yuval Fledel、 Yura Pakhuchiy、 および Cristian Klein が作成しました。

名前

ntfslabel – NTFS ファイルシステムのラベルの表示または変更

形式

`ntfslabel [options] device [new_label]`

機能説明

`ntfslabel` ユーティリティーは、*device* (「形式」を参照) 上にある NTFS ファイルシステムのファイルシステムラベルを表示または変更します。

省略可能な引数 *new_label* を指定しない場合、`ntfslabel` は、現在のファイルシステムのラベルを表示します。

省略可能な引数 *new_label* を指定した場合、`ntfslabel` は、ファイルシステムのラベルを *new_label* に設定します。NTFS ファイルシステムのラベルには、最大で 128 個の Unicode 文字を使用できます。*new_label* が 128 個の Unicode 文字を超えている場合、`ntfslabel` は制限を超えている文字を切り捨て、警告メッセージを表示します。

ファイルシステムの作成中に、`mkntfs(1M)` の `-L` オプションを使用してファイルシステムのラベルを設定することもできます。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-f`、`--force`
: マウントされたボリュームを扱わないなど、一部の妥当なデフォルト値を無効にします。このオプションは慎重に使用してください。

`-h`、`--help`
: それぞれの簡単な説明が付いたオプションのリストを表示します。

`-n`、`--no-action`
: 実際には新しいラベルをディスクに書き込みません。

`-q`、`--quiet`
: 一部のデバッグ、警告、およびエラーメッセージを抑制します。

`-V`、`--version`
: バージョン番号、著作権、およびライセンス情報を表示します。

`-v`、`--verbose`
: 追加のデバッグ、警告、およびエラーメッセージを表示します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

mkntfs(1M), ntfsprogs(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfslabel は、Anton Altaparmakov と Richard Russon の協力の下、Matthew J. Fanto が作成しました。

名前

ntfsls – NTFS ファイルシステムのディレクトリ内容の一覧表示

形式

`ntfsls [options] device`

機能説明

`ntfsls` ユーティリティーは、`PATH` オプション (デフォルトはルートディレクトリ) で指定したファイルに関する情報を一覧表示します。*device* (「形式」を参照) は、デバイスに対応する特殊ファイル (`/dev/dsk/c0d0p0` *num* など)、または NTFS イメージファイルです。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、`-fv` は `-f -v` と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

`-a`、`--all`
: すべてのファイルを表示します。このオプションを指定しない場合、POSIX 名前空間のファイル名は表示されません。

`-F`、`--classify`
: 次に示すインジケータのいずれかをエントリに付加します。

  `* / = @ |`

`-f`、`--force`
: 実行を強制します。たとえば、このオプションは、通常のファイルに格納された NTFS パーティションで実行する場合に必要です。

`-h`、`--help`
: 使用法の情報を表示して終了します。

`-i`、`--inode`
: 各ファイルの i ノード番号を表示します。これは、NTFS 用語での MFT 参照番号です。

`-l`、`--long`
: 長い出力形式を使用します。

`-p`、`--path` *path*
: 内容を一覧表示するディレクトリ、または情報を表示するファイル (パスを含む)。

`-q`、`--quiet`
: 一部のデバッグ、警告、およびエラーメッセージを抑制します。

`-R`、`--recursive`
: 指定したディレクトリの下位にあるすべてのディレクトリの内容を表示します。

`-s`、`--system`
: ドル記号文字で始まるすべてのファイルは通常システムファイルであるため、このオプションを指定しないかぎり、これらのファイルは一覧表示されません。

-v、--verbose
追加のデバッグ、警告、およびエラーメッセージを表示します。

-V、--version
ntfsls のバージョン番号を表示して終了します。

-x、--dos
長いファイル名 (つまり、WIN32 名前空間のファイル) ではなく、短いファイル名 (つまり、DOS 名前空間でのファイル) で表示します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

ntfsprogs(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

このバージョンの ntfsls は、Lode Leroy、Anton Altaparmakov、Richard Russon、Carmelo Kintana、および Giang Nguyen が作成しました。

名前

ntfsprogs – NTFS ツールのリスト

形式

`ntfsprogs`

機能説明

`ntfsprogs` とは、共有ライブラリに基づいた NTFS ユーティリティースイートの名前です。これらのツールは無料で利用でき、完全なソースコードが付属しています。次にこれらのツールを示します。

`mkntfs(1M)`
: NTFS ファイルシステムを作成します。

`ntfscat(1M)`
: ファイルの内容を標準出力にダンプします。

`ntfsclone(1M)`
: NTFS のクローン作成、バックアップ作成、復元、またはレスキューを効率的に行います。

`ntfscluster(1M)`
: 指定したセクターまたはクラスタを使用するファイルを見つけます。

`ntfscmp(1M)`
: 2 つの NTFS ファイルシステムを比較して、相違点を報告します。

`ntfscp(1M)`
: NTFS ボリュームにファイルをコピーします。

`ntfsfix(1M)`
: 一般的なエラーのいくつかを調べて修復し、ログファイルをクリアし、オペレーティングシステムが次回起動したときに徹底的に検査します。

`ntfsinfo(1M)`
: NTFS に関する情報や、その中にあるいずれかのファイルまたはディレクトリに関する情報を表示します。

`ntfslabel(1M)`
: NTFS ファイルシステムのボリュームラベルを表示または設定します。

`ntfsls(1M)`
: NTFS に存在するディレクトリ内のファイルに関する情報を一覧表示します。

`ntfsmount(1M)` (SunOS のマニュアルページではありません)
: NTFS ユーザー空間ドライバを読み書きします。

`ntfsresize(1M)`
: データを失うことなく NTFS のサイズを変更します。

`ntfsundelete(1M)`
: NTFS から削除したファイルを回復します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

これらのツールは、Anton Altaparmakov、Carmelo Kintana、Cristian Klein、Erik Sornes、Giang Nguyen、Holger Ohmacht、Lode Leroy、Matthew J. Fanto、Per Olofsson、Richard Russon、Szabolcs Szakacsits、Yura Pakhuchiy、および Yuval Fledel が作成しました。

名前

ntfsresize – データ損失を伴わない NTFS ファイルシステムのサイズ変更

形式

`ntfsresize [options] --info device`

`ntfsresize [options] [--size size[k|M|G]] device`

機能説明

`ntfsresize` プログラムは、Windows XP、Windows Server 2003、Windows 2000、Windows NT4、および Longhorn NTFS ファイルシステムのサイズを安全に、データを損失することなく変更します。32 ビットおよび 64 ビットの Windows「オペレーティングシステム」で使用されるすべての NTFS バージョンがサポートされます。`ntfsresize` は、データの整合性を危険にさらすことなく、必要に応じてあらゆるデータを再配置できるので、サイズ変更の前にデフラグを行う必要はありません。

マウントされていないデバイス (通常はディスクパーティション) に配置された NTFS ファイルシステムの縮小または拡大に、`ntfsresize` を使用できます。新しいファイルシステムは、指定したサイズになります。サイズパラメータには、オプションの修飾子 `k`、`M`、`G` (それぞれ K バイト、M バイト、G バイトを示す) のいずれかを指定できます。`ntfsresize` は、`k=10`^{3}、`M=10`^{6}、および `G=10`^{9} をサポートすることによって、SI、ATA、IEEE の各標準とディスク製造元に準拠します。

`---info` と `---size` の両方のオプションを指定しない場合、NTFS ファイルシステムは、基礎となるデバイスのサイズに拡大されます。

パーティション上のファイルシステムのサイズを変更するには、ディスク上のパーティションテーブルを編集して、ファイルシステムとパーティションの両方のサイズを変更する必要があります。ほかのコマンド行ファイルシステムサイズ変更ツールと同様に、`ntfsresize` は、パーティションのサイズを操作しません。これを行うには、`fdisk(1M)` などのディスクパーティション分割ツールを使用する必要があります。または、`ntfsresize` を内部で使用する、その他多数のユーザーフレンドリなパーティション分割ツールを使用することもできます。このようなパーティション分割ツールとしては、たとえば Mandriva の DiskDrake、QTParted、SUSE/Novell の YaST Partitioner、IBM の EVMS、GParted、Debian/Ubuntu の Partman があります。

パーティション分割ツールを使用する前に、データとパーティションテーブルのバックアップを作成してください。NTFS ファイルシステムの場合、バックアップの手段として、`ntfsclone(1M)` を使用できます。

NTFS パーティションを縮小するには、最初に `ntfsresize` を使用してファイルシステムのサイズを縮小します。次に、`fdisk(1M)` などのユーティリティーを使用して、パーティションを削除してからより小さなサイズで再作成することによって、パーティションのサイズを縮小します。パーティションのサイズを NTFS の新しいサイズよりも小さくしないでください。小さくすると、そのパーティションから起動できなくなります。間違ってパーティションのサイズを小さくしすぎた場合は、新しいサイズに変更した NTFS ファイルシステムと同じ大きさになるように、パーティションを再作成する必要があります。

NTFS ファイルシステムを拡大するには、最初に、基礎となるパーティションのサイズを拡大する必要があります。fdisk(1M) を使用すると、パーティションを削除して、より大きなサイズで再作成できます。新しいサイズに変更したパーティションが、ほかのどのパーティションとも重なっていないことを確認してください。続いて ntfsresize を使用してファイルシステムを拡大します。

パーティションを再作成するときには、必ず、元のパーティションで使用されたものと同じ開始セクターとパーティションタイプを使用してください。それ以外の場合は、ファイルシステムにアクセスできなくなります。デフォルトのシリンダ単位から信頼できるセクター単位に切り換えるには、fdisk u コマンドを使用します。また、古いパーティションで起動可能フラグが設定されていた場合は、必ず、再作成したパーティションでもこのフラグを設定してください。それ以外の場合、新しいパーティションから起動できなくなる可能性があります。

拡張機能説明

ntfsresize の使用には、非常にまれにしか見られない制限がわずかにあります。このような制限は、たとえば、不明な不良セクターを含むディスクにファイルシステムが格納された場合に生じます。最初の MFT エクステントの再配置と、$MFTMirr エクステントの中央へのサイズ変更はサポートされていません。これらの場合は検出され、サイズ変更は安全なサイズに制限されるか、もっとも近い安全なサイズが表示されます。

サイズ変更が完了すると、ntfsresize は NTFS 整合性検査のスケジュールを設定します。Windows では、この検査は chkdsk によって実行されます。その後、最初に Windows にリブートしたときに、chkdsk が青色のバックグラウンドで実行されます。これは正常な状態です。整合性検査のすぐあとに、Windows により強制的にリブートされる場合があります。ハードウェア構成に応じて、Windows でシステム設定の変更に関する警告が表示され、リブートが推奨または要求されます。メッセージを承認し、もう一度リブートします。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、-fv は -f -v と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

-b、--bad-sectors

デフォルトでは、不良セクターを検出すると、ntfsresize は終了します。このオプションを使用すると、不良セクターがあってもユーティリティーを続行できます。

このオプションを使用する前に、--rescue オプションを指定した ntfsclone(1M) を使用してバックアップを作成したあと、Windows のコマンド行から chkdsk /f /r volume: (volume はボリューム) を実行することをお勧めします。ディスク保証が有効と表示された場合、欠陥があるので交換してください。不良セクターエラーを修復できるソフトウェアは存在しません。これらの欠陥を回避することが、最善の方法です。

ディスクに不良セクターがない場合、このオプションは効果がありません。

`-f`、`--force`
: `ntfsresize` は常に、実際の (`--no-action` を使用しない) サイズ変更操作の前に、整合性検査の対象としてファイルシステムにマークを付け、安全性を高めるためそのままにします。したがって、`ntfsresize` によって NTFS ファイルシステムにマークが付けられた場合は、このオプションを使用しても安全です。サイズ変更の各手順間で Windows を起動せずに複数回サイズを変更する場合は、このオプションを使用する必要があります。

`-h`、`--help`
: 使用法の情報を表示して終了します。

`-i`、`--info`
: ファイルシステムを縮小する必要があるときに使用します。`ntfsresize` が、縮小対象のファイルシステムでサポートされる最小サイズを決定します。ほとんどの場合、最小サイズは、ファイルシステムですでに使用されている容量になります。`ntfsresize` は、このオプションで返されるサイズ未満にファイルシステムを縮小しません。複数の要因により、この理論的なサイズまで縮小できない場合もあります。データの整合性が危険にさらされることはありませんが、それでも、実際のサイズ変更の前に `--no-action` オプションを使用してテスト実行を行うことを強くお勧めします。

  テストによると、最小サイズはおおよそで、ファイルシステムで使用されている容量に 20 - 200M バイトを加えたサイズになります。Windows では、安全に起動するために、さらに 50 - 100M バイトが必要になる可能性があります。

  このオプションを使用しても、ファイルシステムに変更はありません。パーティションは読み取り専用で開かれます。

`-n`、`--no-action`
: サイズ変更操作を行う前にテストを実行する場合に、このオプションを使用します。ボリュームは読み取り専用で開かれます。`ntfsresize` は、ファイルシステムのサイズを変更した場合にどのような影響があるかを表示します。テスト実行に成功した場合にのみ、実際のサイズ変更を続行します。

`-P`、`--no-progress-bar`
: `ntfsresize` の操作中に進捗状況を表示しません。

`-s`、`--size` *size*[k|M|G]
: ファイルシステムのサイズを *size* バイトに変更します。新しいファイルシステムは、指定したサイズになります。サイズパラメータには、オプションの修飾子 `k`、`M`、`G` (それぞれ K バイト、M バイト、G バイトを示す) のいずれかを指定できます。`ntfsresize` は、`k=10`^{3}、`M=10`^{6}、および `G=10`^{9} をサポートすることによって、SI、ATA、IEEE の各標準とディスク製造元に準拠します。実際のサイズ変更を実行する前に、最初に、このオプションとともに `--no-action` オプションを付けて、`ntfsresize` を実行します。

`-v`、`--verbose`
: 多量の出力を表示します。

-V、--version
ntfsresize のバージョン番号を表示します。

終了ステータス

成功した場合は 0 を、失敗した場合は 0 以外を表示します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

fdisk(1M), ntfsclone(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

注意事項

Linux Version 2.6 では、parted(1M) で操作されたパーティションで ntfsresize を使用すると、パーティションテーブルが破壊され、その結果、Windows システムが起動不能になりました。これは、NTFS ファイルシステムに整合性がある場合でも起きていました。この問題は NTFS とは無関係であり、したがって ntfsresize に関係したものではありません。さらに、ntfsresize は決してパーティションテーブルを操作しません。直前に説明した条件では、BIOS でディスクアクセスモードを LBA に変更すれば、起動機能を回復できます。この条件に関する詳細は、http://mlf.linux.rulez.org/mlf/ezaz/ntfsresize.html の ntfsresize の FAQ を参照してください。

著者

ntfsresize は、Anton Altaparmakov と Richard Russon の協力の下、Szabolcs Szakacsits が作成しました。

名前

ntfsundelete – NTFS ボリュームからの削除されたファイルの回復

形式

`ntfsundelete [options] device`

機能説明

`ntfsundelete` ユーティリティーは、適切な状況下で、削除されたファイルを NTFS ボリュームから回復できます。このコマンドには 3 つの操作モードがあります。

`scan`
: デフォルトモードです。`scan` は、NTFS ボリュームを読み取り、削除されたファイルを検索するだけです。検索後、削除された各ファイルの i ノード番号、名前、サイズを一覧表示します。

`undelete`
: undelete モードは、正規表現に一致するファイル (オプション `-m`)、または *inode-expressions* で指定されたファイルのどちらかを取り、できるだけ多くのデータを回復します。別の場所に結果を保存します。

`copy`
: 上級者向けオプションです。MFT の一部をファイルに保存します。このファイルは `ntfsundelete` のデバッグ時に役立ちます。

`ntfsundelete` でファイルを回復できない状況は多数あります。たとえば、次のシナリオを考えます。ファイルを削除したとき、MFT レコードには未使用というマークが付けられ、ディスクの使用状況を表すビットマップが更新されます。電源をすぐに切らない場合、これまでファイルが存在していた空き領域が上書きされる可能性があります。さらに悪い場合は、MFT レコードが別のファイルに再使用されることもあります。このような事態になった場合、ファイルがディスク上のどこにあったかを判別することは不可能です。

ファイルのすべてのクラスタが未使用の場合でも、一部の短命ファイルによって上書きされていないという保証はありません。

`ntfsundelete` は、圧縮または暗号化されたファイルを回復できません。スキャン中、このようなファイルは 0% 回復可能であると表示されます。

ロケール

NTFS では、すべてのファイル名が Unicode で格納されます。ファイル名は、`ntfsundelete` によって現在のロケールに変換されて表示されます。このユーティリティーは漢字ファイル名を正しく表示し、正しく回復します。

拡張 MFT レコード

まれに、ファイルを記述したメタデータを保持するために、単一の MFT レコードでは不足する場合があります (1 つのファイルが数百のフラグメントに分散している場合に不足します)。このような場合は、ある MFT レコードにファイル名が格納され、別のレコードにデータに関する情報が格納される可能性があります。`ntfsundelete` は、これらのレコードをまとめようとはしません。単に、データが含まれる、名前のないファイルを一覧表示するだけです。

回復されたファイルのサイズおよび作成日

ntfsundelete は、ファイルを回復するために、ファイルのメタデータを読み取る必要があります。残念ながら、ファイルが削除されると、メタデータは整合性のない状態になる可能性があります。たとえば、ファイルサイズが 0 と記録される場合や、ファイルの作成日が、削除された日時や無作為な日時に設定される場合があります。これらの状況では、ntfsundelete は、見つかった最大のファイルサイズを選択して、それをディスクに書き込みます。また、ファイルの作成日を最終更新日に設定しようとします。この日付は、正しい最終更新日の場合もあれば、予期しない日付の場合もあります。

オプション

サポートされているオプションを次に示します。ほとんどのオプションには、単一文字と完全名の両方の形式があります。引数を取らない単一文字オプションは複数を組み合わせることができます。たとえば、-fv は -f -v と同等です。完全名のオプションは、その名前の一意の接頭辞に短縮できます。

-b、--byte *num*
回復不可能なファイルクラスタの部分を、*num* で表されるバイトで埋めます。デフォルトは 0 です。

-C、--case
--match オプションを付けてファイル名検索を行うと、照合を行うときに大文字と小文字を区別します。デフォルトのファイル名検索では、大文字と小文字を区別しません。

-c、--copy *range*
この上級者向けオプションは、MFT FILE レコードのブロックをファイルに書き込みます。デフォルトファイルは、現在のディレクトリに作成される mft です。このオプションは、--output オプションおよび --destination オプションと同時に指定できます。

-d、--destination *dir*
--copy オプションと --undelete オプションの出力ファイルの場所を指定します。

-f、--force
既存のファイルを上書きしないなど、一部の妥当なデフォルト値を無効にします。このオプションは慎重に使用してください。

-h、--help
それぞれの簡単な説明が付いたオプションのリストを表示します。

-i、--inodes *range*
指定した範囲の i ノード番号内のファイルを回復します。*range* には、単一の i ノード番号、コンマ区切りの複数の番号、またはダッシュ (-) 区切りの範囲を指定できます。

-m、--match *pattern*
pattern に一致するファイル名だけを検索することによって、出力をフィルタ処理します。パターンには、ちょうど 1 文字に一致するワイルドカード ? や、0 個

以上の文字に一致する * を含めることができます。デフォルトでは、照合では大文字と小文字を区別しません。検索で大文字と小文字を区別するには、--case オプションを使用します。

-O、--optimistic
現在使用中とマークされている場合でも、ファイルの一部を回復します。

-o、--output *file*
--copy オプションまたは --undelete オプションで作成される出力ファイルの名前を設定します。

-P、--parent
削除されたファイルの親ディレクトリを表示します。

-p、--percentage *num*
回復可能率が *num* パーセントの内容のファイルだけに一致するように、--scan オプションの出力をフィルタ処理します。

-q、--quiet
出力の量を最小限にします。このオプションは、--scan オプションを使用する場合には役立ちません。

-s、--scan
NTFS ボリュームを検索して、回復できる可能性があるファイルのリストを表示します。これは ntfsundelete のデフォルトの動作です。このリストは、ファイル名、サイズ、回復可能率、または最終更新時刻でフィルタ処理できます。それぞれ --match、--size、--percent、--time オプションを使用します。

このオプションの出力の %age (パーセント) フィールドには、ファイルの中で回復できる見込みのある割合が表示されます。

-S、--size *range*
特定の範囲のファイルサイズを検索することによって、--scan オプションの出力をフィルタ処理します。*range* は、2 つの数値をハイフン (-) で区切って指定できます。サイズの単位は、K バイト、M バイト、G バイト、T バイトに対してそれぞれ接尾辞 k、m、g、t の略称を使用できます。

-t、--time *since*
--scan オプションの出力をフィルタ処理します。この時間以降に変更されたファイルだけを照合します。時間は、数値と、日、週、月、年にそれぞれ対応する接尾辞 d、w、m、y で指定する必要があります。

-T、--truncate
ntfsundelete のデフォルトの動作では、ファイルのサイズをもっとも近いクラスタに切り上げます (512 バイトの倍数になります)。ユーティリティーにファイルのサイズに関する完全なデータがある場合、このオプションは正確にそのサイズにファイルを復元します。

-u、--undelete
undelete モードを指定します。--match オプションや --inodes オプションを使用することにより、回復対象のファイルを指定できます。このオプションは、--output、--destination、および --byte と同時に指定できます。

--output オプションを使用しないかぎり、ファイルを回復するときにその元の名前が付けられます。

-v、--verbose
ntfsundelete が表示する出力の量を増やします。

-V、--version
ntfsundelete のバージョン番号、著作権、およびライセンスを表示します。

使用例

例 1 削除されたファイルを検索する

次のコマンドは、特定のデバイス上で削除されたファイルを検索します。

```
# ntfsundelete /dev/dsk/c0d0p1
```

例 2 ワイルドカードに一致するファイルを検索する

次のコマンドは、*.doc に一致する削除されたファイルを検索します。

```
# ntfsundelete /dev/dsk/c0d0p1 -s -m '*.doc'
```

例 3 特定のサイズのファイルを検索する

次のコマンドは、少なくとも 90% のデータが回復可能な 5000 から 6000000 バイトの削除されたファイルを、/dev/dsk/c0d0p1 上で検索します。

```
# ntfsundelete /dev/dsk/c0d0p1 -S 5k-6m -p 90
```

例 4 最近変更されたファイルを検索する

次のコマンドは、削除されたファイルのうち過去 2 日間に変更されたものを検索します。

```
# ntfsundelete /dev/dsk/c0d0p1 -t 2d
```

例 5 i ノード範囲を指定する

次のコマンドは、デバイス /dev/sda1 の 2、5、および 100 から 131 の i ノードの削除を取り消します。

```
# ntfsundelete /dev/sda1 -u -i 2,5,100-131
```

例 6 出力ファイルとディレクトリを指定する

次のコマンドは、i ノード番号 3689 の削除を取り消し、ファイルに work.doc の名前を付け、ユーザーのホームディレクトリにそのファイルを格納します。

```
# ntfsundelete /dev/dsk/c0d0p1 -u -i 3689 -o work.doc -d ~
```

例 7　MFT レコードを保存する

次のコマンドは、MFT レコード 3689 から 3690 を、ファイル debug に保存します。

```
# ntfsundelete /dev/dsk/c0d0p1 -c 3689-3690 -o debug
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/ntfsprogs
インタフェースの安定性	不確実

関連項目

ntfsclone(1M), ntfsresize(1M), parted(1M), attributes(5)

http://wiki.linux-ntfs.org

著者

ntfsundelete は、Anton Altaparmakov の協力の下、Richard Russon および Holger Ohmacht が作成しました。

名前

nwamd – network auto-magic デーモン

形式

/lib/inet/nwamd

機能説明

nwamd は、ネットワークインタフェースを管理するシステムデーモンです。

このデーモンは、network/physical:default サービスによって自動的に起動されるので、直接起動しないようにしてください。これは、プログラミングインタフェースの構成要素ではありません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	流動的

関連項目

svcs(1), netcfgd(1M), netadm(1M), netcfg(1M), svcadm(1M), attributes(5), nwam(5), smf(5)

JDS/GNOME マニュアルページコレクションで利用可能な nwam-manager(1M) も参照してください。

注意事項

ネットワークサービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/network/physical:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

obpsym – Kernel Symbolic Debugging for OpenBoot Firmware

形式

```
modload -p misc/obpsym
```

機能説明

`obpsym` is a kernel module that installs OpenBoot callback handlers that provide kernel symbol information to OpenBoot. OpenBoot firmware user interface commands use the callbacks to convert numeric *addresses* to kernel symbol names for display purposes, and to convert kernel symbol names to numeric *literals* allowing symbolic names to be used as input arguments to user interface commands.

Once `obpsym` is installed, kernel symbolic names may be used anywhere at the OpenBoot firmware's user interface command prompt in place of a literal (numeric) string. For example, if `obpsym` is installed, the OpenBoot firmware commands `ctrace` and `dis` typically display symbolic names and offsets in the form *modname:symbolname* + *offset*. User interface Commands such as `dis` can be given a kernel symbolic name such as `ufs:ufs_mount` instead of a numeric address.

Placing the command

```
forceload: misc/obpsym
```

into the `system(4)` file forces the kernel module `misc/obpsym` to be loaded and activates the kernel callbacks during the kernel startup sequence.

`obpsym` may be useful as a kernel debugger in situations where other kernel debuggers are not useful. For example, on SPARC machines, if `obpsym` is loaded, you may be able to use the OpenBoot firmware's `ctrace` command to display symbolic names in the stack backtrace after a watchdog reset.

Kernel Symbolic Name Syntax

The syntax for a kernel symbolic name is:

[*module-name* :] *symbol-name*

Where *module-name* is the name of the kernel module that the symbol *symbol-name* appears in. A `NULL` module name is taken as “all modules, in no particular order” by `obpsym`. The module name `unix` is equivalent to a `NULL` module name, so that conflicts with words defined in the firmware's vocabulary can be avoided.

Typically, OpenBoot firmware reads a word from the input stream and looks the word up in its internal *vocabulary* before checking if the word is a *literal*. Thus, kernel symbols, such as `reset` may be given as `unix:reset` to avoid the unexpected side effect of the firmware finding and executing a matching word in its vocabulary.

ファイル

`/etc/system`	System configuration information file.
`/platform/`*platform-name*`/kernel/misc/obpsym`	Platform-specific kernel symbol information.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/processor

関連項目

kmdb(1), uname(1), kernel(1M), modload(1M), modunload(1M), system(4), attributes(5)

『OpenBoot 2.x コマンド・リファレンスマニュアル』

警告

Some OpenBoot firmware user interface commands may use system resources incompatibly with the way they are used by the Unix kernel. These commands and the use of this feature as a kernel debugger may cause interactions that the Unix kernel is not prepared to deal with. If this occurs, the Unix kernel and/or the OpenBoot firmware user interface commands may react unpredictably and may panic the system, or may hang or may cause other unpredictable results. For these reasons, the use of this feature is only minimally supported and recommended to be used only as a kernel debugger of last resort.

If a breakpoint or watchpoint is triggered while the console frame buffer is powered off, the system can crash and be left in a state from which it is difficult to recover. If one of these is triggered while the monitor is powered off, you will not be able to see the debugger output.

注意事項

platform-name can be found using the `-i` option of uname(1)

`obpsym` is supported only on architectures that support OpenBoot firmware.

On some systems, OpenBoot must be completely RAM resident so the `obpsym` symbol callback support can be added to the firmware, if the firmware doesn't include support for the symbol callbacks. On these systems, `obpsym` may complain that it requires that “you must use `ramforth` to use this module”.

See the 『OpenBoot 2.x コマンド・リファレンスマニュアル』 for details on how to use the *ramforth* command, how to place the command into *nvramrc*, and how to set *use-nvramrc?* to `true`. On systems with version 1.x OpenBoot firmware, *nvramrc* doesn't exist, and the *ramforth* command must be typed manually after each reset, in order to use this module.

Once installed, the symbol table callbacks can be disabled by using the following OpenBoot firmware command:

```
0 0 set-symbol-lookup
```

名前

oplhpd – Hot plug daemon for SPARC Enterprise Server line

形式

`/usr/platform/SUNW,SPARC-Enterprise/lib/sparcv9/lib/oplhpd`

機能説明

The hot plug daemon for SPARC Enterprise Servers is a daemon process that runs on the SUNW,SPARC-Enterprise family of servers. The daemon is started by the service management facility (see `smf(5)`) and communicates with the service processor when hot plug PCI cassettes change their dynamic reconfiguration state.

The service FMRI for `oplhpd` is:

```
svc:/platform/sun4u/oplhpd:default
```

A domain supports only one running `oplhpd` process at a time.

エラー

OPLHPD uses `syslog(3C)` to report status and error messages. All of the messages are logged with the LOG_DAEMON facility.

Error messages are logged with the LOG_ERR and LOG_NOTICE priorities, and informational messages are logged with the LOG_DEBUG priority. The default entries in the `/etc/syslog.conf` file log all of the OPLHPD error messages to the `/var/adm/messages log`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/domain-configuration/sparc-enterprise, SUNWdcsr
Interface Stability	Committed

関連項目

`svcs(1)`, `inetadm(1M)`, `svcadm(1M)`, `syslog(3C)`, `syslog.conf(4)`, `attributes(5)`, `smf(5)`

名前

parted – パーティション操作プログラム

形式

`parted [options] [device [command [options...]...]]`

機能説明

`parted` は、ディスクパーティション分割およびパーティションサイズ変更プログラムです。`ext2`、`linux-swap`、`FAT`、`FAT32`、および `reiserfs` パーティションの作成、破棄、サイズ変更、移動、およびコピーを行えます。Macintosh HFS パーティションの作成、サイズ変更、および移動のほか、`jfs`、`ntfs`、`ufs`、および `xfs` パーティションの検出も行えます。新しいオペレーティングシステム用の領域の作成、ディスク使用法の再編成、および新しいハードディスクへのデータのコピーに役立ちます。

このマニュアルページでは `parted` について簡単に説明します。`parted` に関する完全なドキュメントは、「GNU Info」形式のパッケージで配布されています。

`parted` には、一連の最上位オプションとサブコマンドが実装され、それらのほとんどは独自のオプションとオペランドを備えています。次に、これらのサブコマンドについて説明します。`parted` にはオプションのオペランドがあります。

device 使用されるブロックデバイス。指定されない場合、`parted` は、最初に見つかったブロックデバイスを使用します。

引数を付けずに `parted` を呼び出した場合、プログラムはコマンドプロンプトを表示します。

オプション

次のオプションがサポートされています。

`-a` *alignment-type*、`--align` *alignment-type*

新しく作成したパーティションのアラインメントを設定します。有効なアラインメントタイプは次のとおりです。

`none`

ディスクタイプで使用できる最小のアラインメントを使用します。

`cylinder`

パーティションをシリンダに合わせて調整します。

`minimal`

ディスクトポロジ情報から得られる最小のアラインメントを使用します。この値と *optimal* 値は、ディスクから得られたレイアウト情報を使用して、論理的なパーティションテーブルアドレスを、ディスク上の実際の物理的なブロックに合わせて調整します。*minimal* 値は、パーティションを物理的ブロックに合わせて適切に調整するために必要な最小のアラインメントであり、パフォーマンスの低下を回避します。

`optimal`

ディスクトポロジ情報から得られる最適なアラインメントを使用します。これは、最適なパフォーマンスが確保されるように、物理的なブロックサイズの倍数に合わせて調整します。

`-h`、`--help`
ヘルプメッセージを表示します。

`-l`、`--list`
すべてのブロックデバイスのパーティションレイアウトを一覧表示します。

`-m`、`--machine`
マシン解析可能な出力を表示します。

`-s`、`--script`
ユーザーの操作を要求しません。

`-v`、`--version`
バージョン番号を表示します。

サブコマンド

`parted` コマンド行でサブコマンドを省略した場合、ユーティリティーはコマンドプロンプトを発行します。

`check` *partition*
partition を簡単にチェックします。

`cp` [*source-device*] *source dest*
source-device (または、ほかのデバイスが指定されていない場合は現在のデバイス) 上のソースパーティションのファイルシステムを、現在のデバイス上の *dest* パーティションにコピーします。

`help` *command*
一般ヘルプまたは指定された場合はコマンドに関するヘルプを表示します。

`mkfs` *partition fs-type*
パーティションにファイルシステム *fs-type* を作成します。*fs-type* には、`fat16`、`fat32`、`ext2`、`linux-wap`、`reiserfs` のいずれかを指定できます。

`mklabel` *label-type*
label-type の新しいディスクラベル (パーティションテーブル) を作成します。*label-type* には、`bsd`、`dvh`、`gpt`、`loop`、`mac`、`msdos`、`pc98`、`sun` のいずれかを指定するようにしてください。

`mkpart` *part-type* [*fs-type*] *start end*
ファイルシステム *fs-type* (指定した場合) の、*start* から始まり、*end* で終わる (デフォルトで M バイト単位) *part-type* パーティションを作成します。*fs-type* には、`fat16`、`fat32`、`ext2`、`HFS`、`linux-swap`、`NTFS`、`reiserfs`、`ufs` のいずれかを指定できます。*part-type* には、`primary`、`logical`、`extended` のいずれかを指定するようにしてください。

`mkpartfs` *part-type fs-type start end*
ファイルシステム *fs-type* の、*start* から始まり、*end* で終わる (デフォルトで M バイト単位) *part-type* パーティションを作成します。

このサブコマンドの使用は推奨されません。mkpart を使用して空のパーティションを作成する代わりに、mke2fs(8) (Linux の一部) などの外部ツールを使用してファイルシステムを作成してください。

move *partition start end*
start から始まり *end* で終わるように、パーティションを移動します。move ではマイナー番号が決して変更されません。

name *partition name*
パーティションの名前を *name* に設定します。このオプションは、Mac、PC98、および GPT のディスクラベルでのみ機能します。名前は必要に応じて引用符で囲むことができます。

print
パーティションテーブルを表示します。

quit
parted を終了します。

rescue *start end*
start と *end* の間のどこかの地点に存在したが失われたパーティションをレスキューします。パーティションが見つかった場合、parted は、パーティションテーブル内にそのパーティション用のエントリを作成するかどうかを尋ねます。

resize *partition start end*
start から始まり *end* で終わるように、*partition* 上のファイルシステムのサイズを変更します (デフォルトで M バイト単位)。

rm *partition*
partition を削除します。

select *device*
編集対象の現在のデバイスとして *device* を選択します。*device* は、通常、Solaris または Linux ハードディスクデバイスであるべきですが、必要に応じて、パーティション、ソフトウェア RAID デバイス、SVM または LVM 論理ボリュームにすることができます。

set *partition flag state*
partition 上の *flag* の状態を *state* に変更します。サポートされているフラグは、boot、root、swap、hidden、raid、lvm、lba、および palo です。*state* は、on または off のどちらかにするようにしてください。

unit *unit*
場所およびサイズを表示するときに使用する単位として、およびユーザーが指定した値に明示的に単位が接尾辞として付加されていないときにその値を解釈する単位として *unit* を設定します。*unit* は、s (セクター)、B (バイト)、kB、MB、GB、TB、% (デバイスサイズのパーセント)、cyl (シリンダ)、chs (シリンダ、ヘッド、セクター)、compact (入力には M バイト、出力には人間に読みやすい形式) のいずれかにすることができます。

version
バージョン情報と著作権メッセージを表示します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/parted
インタフェースの安定性	不確実

関連項目

fdisk(1M), mkfs(1M), attributes(5)

parted プログラムについては、info(1) 形式の GNU パーティション分割ソフトウェアマニュアルで詳しく説明しています。

注意事項

ext3 ファイルシステム機能は現在動作しません。ext3 タイプのファイルシステムを管理するには、resize2fs(8) や mke2fs(8) (どちらも Linux の一部) などのツールを使用します。現在サポートされている ext2 ファイルシステムは、ext3 サポートの完成後、非推奨になります。ext3 サポートには、まだ定義されていない制限付き機能が含まれています。これらのタイプのファイルシステムを管理するには、resize2fs(8) や mke2fs(8) などのツールを使用します。

ext3 ファイルシステムまたはパーティションのサイズを手動で変更するには、resize2fs(8)、fdisk(8)、または同様のツールを使用します。LVM の場合は、LVM コマンドを使用して LVM 要素のサイズを変更する必要があります。

著者

このマニュアルページは、Timshel Knoll が Debian GNU/Linux システム用に記述しました。ここでは、Solaris オペレーティングシステムに合わせて書き換えられています。

名前

pbind – control and query bindings of processes or LWPs

形式

```
pbind -b processor_id pid [/lwpid]...
pbind [-q] [pid [/lwpid]]...
pbind -Q [processor_id]...
pbind -u pid [/lwpid]...
pbind -U [processor_id]...
```

機能説明

`pbind` controls and queries bindings of processes and LWPs (lightweight processes) to processors. `pbind` can also remove processor bindings that were previously established.

When an LWP is bound to a processor, it will be executed only by that processor except when the LWP requires a resource that is provided only by another processor. The binding is not exclusive, that is, the processor is free to execute other LWPs as well.

Bindings are inherited, so new LWPs and processes created by a bound LWP will have the same binding. Binding an interactive shell to a processor, for example, binds all commands executed by the shell.

Superusers may bind or unbind any process or LWP, while other users can bind or unbind any process or LWP for which they have permission to signal, that is, any process that has the same effective user ID as the user.

オプション

The following options are supported:

`-b` *processor_id*
: Binds all or a subset of the LWPs of the specified processes to the processor *processor_id*. Specify *processor_id* as the processor ID of the processor to be controlled or queried. *processor_id* must be present and on-line. Use the `psrinfo` command to determine whether or not *processor_id* is present and on-line. See `psrinfo(1M)`.

`-q`
: Displays the bindings of the specified processes or of all processes. If a process is composed of multiple LWPs which have different bindings and the LWPs are not explicitly specified, the bindings of only one of the bound LWPs will be displayed. The bindings of a subset of LWPs can be displayed by appending "/lwpids" to the process IDs. Multiple LWPs may be selected using "-" and "," delimiters. See EXAMPLES.

`-Q`
: Displays the LWPs bound to the specified list of processors, or all LWPs with processor bindings. For processes composed of multiple LWPs, the bindings of individual LWPs will be displayed.

`-u`
: Removes the bindings of all or a subset of the LWPs of the specified processes, allowing them to be executed on any on-line processor.

`-U`
Removes the bindings of all LWPs bound to the specified list of processors, or to any processor if no argument is specified.

オペランド

The following operands are supported:

pid
The process ID of the process to be controlled or queried.

lwpid
The set of LWP IDs of the specified process to be controlled or queried. The syntax for selecting LWP IDs is as follows:

```
2,3,4-8       LWP IDs 2, 3, and 4 through 8
-4            LWPs whose IDs are 4 or below
4-            LWPs whose IDs are 4 or above
```

processor_id
The processor ID of the processor to be controlled or queried.

使用例

例 1 Binding Processes

The following example binds processes 204 and 223 to processor 2:

```
example% pbind -b 2 204 223
process id 204: was 2, now 2
process id 223: was 3, now 2
```

例 2 Unbinding a Process

The following example unbinds process 204:

```
example% pbind -u 204
```

例 3 Querying Bindings

The following example queries bindings. It demonstrates that process 1 is bound to processor 0, process 149 has at least one LWP bound to CPU3, and process 101 has no bound LWPs.

```
example% pbind -q 1 149 101
process id 1: 0
process id 149: 3
process id 101: not bound
```

例 4 Querying LWP Bindings

The following example queries bindings of LWPs. It demonstrates that LWP 1 of process 149 is bound to CPU3, and LWP 2 of process 149 is not bound.

```
example% pbind -q 149/1-2
lwp id 149/1: 3
lwp id 149/2: not bound
```

例 5 Querying LWP Bindings for Processor 2:

The following example queries all LWPs bound to processor 2:

```
example% pbind -Q 2
lwp id 149/4: 2
lwp id 149/5: 2
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`>0`
: An error occurred.

関連項目

psradm(1M), psrinfo(1M), psrset(1M), processor_bind(2), processor_info(2), sysconf(3C), attributes(5)

診断

`pbind: cannot query pid 31: No such process`
: The process specified did not exist or has exited.

`pbind: cannot bind pid 31: Not owner`
: The user does not have permission to bind the process.

`pbind: cannot bind pid 31: Invalid argument`
: The specified processor is not on-line.

名前

pcitool – 割り込み経路指定ツール

形式

```
/usr/sbin/pcitool -h
```

x86:

```
/usr/sbin/pcitool pci@unit-address -i cpu#,ino# | all [-r [-c] |
     -w cpu# [-g] ] [-v] [-q]
```

SPARC:

```
/usr/sbin/pcitool pci@unit-address -i ino# | all [-r [-c] |
     -w cpu# [-g] ] [-v] [-q]
/usr/sbin/pcitool pci@unit-address -m msi# | all [-r [-c] |
     -w cpu# [-g] ] [-v] [-q]
```

機能説明

`PCItool` は、割り込み経路指定情報の取得および設定機能を提供する低レベルツールです。

割り込み経路指定

`pcitool -i` コマンドは、指定したネクサス上の INO に関するデバイスおよび CPU 経路指定情報を表示し、指定した INO または INO グループを、指定した CPU へ再経路指定できます。

SPARC プラットフォームでは、INO は割り込みモンドに対応付けられ、1 つまたは複数の MSI/X が 1 つの INO に対応付けられます。したがって、INO と MSI/X は別々に再ターゲット化できます。`-i` オプションを使用して、指定した INO を取得または再経路指定します。MSI/X の場合は `-m` オプションを使用します。

cpu# の指定は、x86 プラットフォーム上で使用できます。*ino#* と組み合わせた場合、これは排他的なベクトルを識別します。*cpu#* 引数は、SPARC プラットフォームではサポートされません。

必要な特権

割り込み情報にアクセスするには、ユーザーはすべての特権を保有している必要があります。一般ユーザーは、`su(1M)` を実行して root になるか、または `user_attr` ファイルで「Maintenance and Repair」権利プロファイルが与えられている場合に、割り込み情報にアクセスできます。`user_attr(4)` および `rbac(5)` を参照してください。

オプション

次のオプションがサポートされています。

`-h`
: コマンドの使用法を表示します。

`-q`
: エラーはメッセージとして表示されません。ただし、`pcitool` は、引き続き UNIX エラーコードを返します。

`-r [-c]`
: 指定したネクサスの INO について、デバイスおよび CPU 経路指定情報を表示します。表示される INO ごとに、各デバイスのデバイスパスとインスタンス番号

が表示されます。一部のプラットフォームでは、ルートコンプレックス専用の割り込みは、そのパス名に付加された文字列 (`Internal`) で示されます。

`-c` を付けると、割り込みコントローラ情報をダンプします。

コマンド行で `-r` も `-w` も指定されない場合、`-r` が指定されたものとみなされます。「使用例」を参照してください。

`-v`
詳細情報を出力します。

`-w` *cpu#* [`-g`]
指定した INO または MSI/X を、指定した CPU に経路指定します。新しい経路指定情報と元の経路指定情報を表示します。INO または MSI/X を指定する必要があります。

一部のプラットフォーム (x86 など) では、単一の機能の複数の MSI 割り込みは一緒に再経路指定する必要があります。これを行うには `-g` を使用します。`-g` オプションは、サポート対象のプラットフォームでのみ、MSI 割り込みのグループにかぎり有効です (1 の「グループ」は受け入れられます)。`-g` を使用する場合、指定するベクトルは、グループの中でもっとも小さい番号のベクトルである必要があります。グループのサイズは内部で決定されます。「使用例」を参照してください。

使用例

例 1　すべての INO を表示する

`/pci@0,0` のすべての INO を表示するコマンドは次のとおりです。

```
# pcitool /pci@0,0 -i all
```

例 2　特定の INO の出力を表示する

ルート `/pci@0,0` の INO `0x0,0x21` を表示するコマンドは、x86 プラットフォームと SPARC プラットフォームでは若干異なります。

x86 プラットフォームの場合

```
# pcitool /pci@0,0 -i 0,21
0x0,0x21:                    mpt                          0
/pci@7b,0/pci1022,7458@11/pci1000,3060@2
```

SPARC プラットフォームの場合

```
# pcitool /pci@0,0 -i 21
0x0,0x21:                    mpt                          0
/pci@7b,0/pci1022,7458@11/pci1000,3060@2
```

前述の出力は一例であり、実際の出力とは異なる場合があります。

例 3　特定の MSI の出力を表示する

ルート `/pci@0,0` の MSI `0x1` を表示するコマンドを、出力例とともに次に示します。

例 3　特定の MSI の出力を表示する　　*(続き)*

```
# pcitool /pci@0,0 -m 0x1
0x0,0x1: pcieb     0    /pci@7b,0/pci10de,5d@e
```

例 4　ある CPU から別の CPU に INO を再経路指定する

CPU 0 から CPU 1 に INO 21 を正しく再経路指定すると、次に示す出力が生成されます。

x86 プラットフォームの場合

```
# pcitool /pci@0,0 -i 0,21 -w 1
0x0,0x21 -> 0x1,0x20
```

SPARC プラットフォームの場合

```
# pcitool /pci@0,0 -i 21 -w 1
0x0,0x21 -> 0x1,0x20
```

例 5　ある CPU から別の CPU に MSI を再経路指定する

CPU 1 から CPU 0 に MSI 1 を正しく再経路指定すると、次に示す出力が生成されます。

```
# pcitool /pci@0,0 -m 1 -w 0
0x1,0x1 -> 0x0,0x1
```

例 6　INO のグループを再経路指定する

CPU 0 から CPU 1 に 24 から始まる INO のグループを正しく再経路指定すると、次に示す出力が生成されます。

x86 プラットフォームの場合

```
# pcitool /pci@0,0 -i 3,24 -w 1 -g
0x3,0x24 => 0x1,0x22
```

SPARC プラットフォームの場合

```
# pcitool /pci@0,0 -i 24 -w 1 -g
0x3,0x24 => 0x1,0x22
```

終了ステータス

`0`
　エラーなし。

`EINVAL`
　範囲外、整列エラー、またはそのほかの無効な引数が渡されました。

`ETIME`
　新しい CPU に割り込みを変更する前に、保留中の割り込みが処理されるのを待機しているときにタイムアウトが発生しました。

EIO
I/O エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	PCI ベースのシステムに限定
使用条件	system/management/pcitool
インタフェースの安定性	流動的

関連項目

su(1M), pci(4), user_attr(4), attributes(5), rbac(5)

PCI 仕様 (www.pcisig.org から入手可能)

注意事項

すべての値は 16 進数で入力します。

すべてのコマンドが全プラットフォームに適用できるわけではありません。

名前

pginfo – プロセッサグループに関する情報の表示

形式

```
pginfo [-T] [-p] [-v] [-r string] [-R string]
     [pg ... | -c processor_id ...]

pginfo -s [-v] [-r string] [-R string] [pg ... | -c processor_id ...]

pginfo -c | -I [-r string] [-R string] [pg ... | -c processor_id ...]

pginfo -h
```

機能説明

pginfo は、プロセッサグループ (PG) 階層、その内容、およびその特性に関する情報を表示します。PG とは、共通の特性でひとまとめにした CPU のセットです。

PG は、オペレーティングシステムによって、実行パイプラインやキャッシュなど、パフォーマンスに関連したハードウェアを共有する CPU を表すために使用されます。これらの PG は、マシンのプロセッサトポロジをモデル化した階層に編成されます。この階層では、それぞれの CPU (ストランド) が、その CPU ともっとも多くのハードウェアを共有している CPU 群を表すリーフ PG を 1 つ持ちます。ルート PG に到達するまでリーフ PG を上位にさかのぼっていくにつれ、その CPU と共有されるハードウェアの数は少なくなっていきます。ルート PG は、システム内のすべての CPU を含み、相互に共有するハードウェアがもっとも少ない CPU のグループを表します (PG 階層の例については、後述の「使用例」を参照してください)。

マシンに、パフォーマンス関連のハードウェアの共有関係がない場合、pginfo は、システム内のすべての CPU を含むルート PG のみを表示します。

デフォルトで、pginfo は、PG ID、共有関係、オンラインおよびオフラインの CPU など、システム内の各 PG に関する情報を表示します。デフォルトで PG を深さ優先順で表示し、PG がどのように相互に関連しているかをインデントを使用してわかりやすく示します (後述の「使用例」を参照)。

オプションを指定して次の操作を行えます。

- PG 階層をグラフィカルに表示する
- 稼働中のシステムに存在する PG 共有関係を一覧表示する
- PG ID、CPU ID、または共有関係で対象の PG を指定して、PG の現在の使用状況情報を表示する
- CPU ID または PG ID だけを表示するように指定する

さらに、system、chip、core などの一般的な物理的関係を持つ CPU に相当する CPU を含む PG を表示する、-p オプションがあります。これらの物理的関係は、CPU の物理的特性を表し、パフォーマンスに関連したプロセッサ共有関係をカプセル化している場合としていない場合があります。

pginfo がシステムデータのスナップショットを取得しているときに、システム構成が繰り返し変更された場合、pginfo はエラーメッセージを表示して、終了ステータス 1 で終了します。

オプション　次のオプションがサポートされています。

-c *processor_id* ...
引数をプロセッサ ID と解釈して、指定されたプロセッサを含む PG に関する情報のみを表示します。

このオプションを -T オプションと一緒に使用すると、PG 階層には、指定された各 CPU の系列のみが表示されます。このオプションは、PG ID で対象の PG を指定する場合には使用できません。

-C
PG に所属するすべての CPU の CPU ID のみを表示します。このオプションは、-I オプションと同時に使用することはできません。

-h
簡単なヘルプメッセージを表示し、終了ステータス 0 で終了します。

-I
PG の PG ID のみを表示します。このオプションは、-C オプションと同時に使用することはできません。

-p
PG に対応する物理的関係を表示します。PG に、システム全体、プロセッサコア、またはチップと同じ CPU がある場合、それぞれ system、core、または chip が、PG の共有関係のあとに角括弧 (「[]」) で囲まれて表示されます。

-r *string1*, *string2*,...
指定された文字列のいずれかに一致した共有関係名を持つ PG に関する情報のみを表示します。

指定する各文字列は、完全な関係名でも、1 つ以上の関係名の一部でも構いません。文字列の照合では大文字と小文字は区別されません。使用可能な関係名は、-s オプションで表示される共有関係のリストに示されます。

複数の -r オプションを指定できますが、この場合、指定された文字列のいずれかが含まれる関係名を持つすべての PG に一致することになります。このオプションを -T オプションと一緒に使用すると、PG 階層には、指定された関係を持つ各 PG の系列のみが表示されます。

-R *string1*, *string2*,...
指定された名前以外の共有関係名を持つ PG に関する情報のみを表示します。

文字列の照合は、前述の -r オプションの説明と同じです。複数の -R オプションを入力できます。

-s
指定された PG について、稼働中のシステムでサポートされるすべての共有関係を表示します。このオプションとともに -v オプションを使用すると、共有関係ごとの PG のリストを取得できます。

-T
この結果表示される階層では、各 CPU (ハードウェアストランド) の系列は、もっとも多くのハードウェアを CPU と共有する PG から、もっとも少ないハードウェアを CPU と共有する PG の順番に並べられます。対象の CPU、PG、または関係が指定されると、結果として表示される PG 階層は、その PG 階層の中で、指定された CPU、PG、または関係を持つ PG の系列に限定されます。

-v
冗長モード。PG に関する追加情報を表示します。-s、-C、または -I を付けずに使用した場合は、-T オプションと -p オプションを同時に指定した場合と同等です。-s オプションを付けて使用した場合は、共有関係ごとの PG のリストが表示されます。

オペランド

1 つまたは複数の対応する ID、またはキーワード `all` を指定することによって、コマンド行で次のオペランドを入力できます。スペース区切りのリスト (たとえば `1 3`)、数値の範囲 (たとえば `5-8`)、またはその両方 (たとえば `1 3 5-8 13-16`) で、複数の ID を指定できます。PG と CPU を同時に指定することはできません。

pg 対象の PG は、コマンド行から PG ID を使って指定できます。

processor_id `-c` オプションを入力した場合、対象の CPU をコマンド行から CPU ID を使って指定できます。

指定された PG または CPU が無効である場合、`pginfo` コマンドは無効な ID を表示するメッセージを標準エラー出力に表示し、コマンド行で指定されたほかの PG または CPU を引き続き処理します。指定された PG または CPU が 1 つも有効でない場合、`pginfo` は終了ステータス 2 で終了します。

使用例

次の例では、システムには、8 つのコアと 32 のストランドを備えた 1 つの UltraSPARC T1 プロセッサチップが搭載されています。

例 1 すべての PG に関する情報を表示する

次の引数なしのコマンドは、すべての PG に関する情報を表示します。

```
$ pginfo
PG  RELATIONSHIP            CPUs
0   System                  0-31
3    Data_Pipe_to_memory    0-31
2     Floating_Point_Unit   0-31
1      Integer_Pipeline     0-3
4      Integer_Pipeline     4-7
5      Integer_Pipeline     8-11
6      Integer_Pipeline     12-15
7      Integer_Pipeline     16-19
8      Integer_Pipeline     20-23
9      Integer_Pipeline     24-27
```

例 1　すべての PG に関する情報を表示する　　*(続き)*

```
10     Integer_Pipeline     28-31
```

例 2　すべての共有関係に関する情報を表示する

次のコマンドは、すべての共有関係に関する情報を表示します。

```
$ pginfo -s -v
RELATIONSHIP         PGs
-------------------- -------
System               0
Data_Pipe_to_memory  3
Floating_Point_Unit  2
Integer_Pipeline     1 4-10
```

例 3　PG 階層を表示する

次のコマンドは、システム内のすべての PG に関する一般情報を表示します。出力には、どの PG がチップおよびコアに所属するかが示されます。

```
$ pginfo -p -T
0 (System) CPUs: 0-31
`-- 3 (Data_Pipe_to_memory [system,chip]) CPUs: 0-31
    `-- 2 (Floating_Point_Unit [system,chip]) CPUs: 0-31
        |-- 1 (Integer_Pipeline [core]) CPUs: 0-3
        |-- 4 (Integer_Pipeline [core]) CPUs: 4-7
        |-- 5 (Integer_Pipeline [core]) CPUs: 8-11
        |-- 6 (Integer_Pipeline [core]) CPUs: 12-15
        |-- 7 (Integer_Pipeline [core]) CPUs: 16-19
        |-- 8 (Integer_Pipeline [core]) CPUs: 20-23
        |-- 9 (Integer_Pipeline [core]) CPUs: 24-27
        `-- 10 (Integer_Pipeline [core]) CPUs: 28-31
```

例 4　特定の基準でリストを表示する

次のコマンドは、CPU 0 と整数パイプラインを共有する CPU のリストを表示します。この例ではさらに、-r オプションを使用して PG を共有関係名でフィルタ処理しています。

```
$ pginfo -r integer_pipeline -C -c 0
0 1 2 3
```

例 5　特定の基準で排除するオプションを使用する

次のコマンドは、`Integer_Pipeline` の関係を持つ PG 以外のすべての PG を一覧表示します。

```
$ pginfo -R Integer_Pipeline
PG RELATIONSHIP            CPUs
```

例 5 特定の基準で排除するオプションを使用する (続き)

```
0  System                  0-31
3  Data_Pipe_to_memory     0-31
2  Floating_Point_Unit     0-31
```

終了ステータス

0
正常終了。

1
エラーが発生しました。

2
構文が正しくありません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	不確実

コマンド行オプションおよび出力は「不確実」です。

関連項目

pgstat(1M), attributes(5)

名前

pgstat – プロセッサグループの使用状況統計の報告

形式

```
pgstat [-A] [-C] [-p] [-s key | -S key] [-t number] [-T u | d] [-v]
[-r string] [-R string] [-P pg ...] [-c processor_id... ]
[interval [count]]

pgstat -h
```

機能説明

pgstat はプロセッサグループ (PG) に関する使用状況統計を表示します。PG とは、共通の特性でひとまとめにした CPU のセットです。

PG は、オペレーティングシステムで、実行パイプラインやキャッシュなど、パフォーマンスに関連したハードウェアを共有する CPU を表すために使用されます。これらの PG は、マシンのプロセッサトポロジをモデル化した階層に編成されます。この階層では、それぞれの CPU (ストランド) に、その CPU ともっとも多くのハードウェアを共有している CPU を表すリーフ PG が割り当てられます。ルート PG に到達するまでリーフ PG を上位にさかのぼっていくにつれ、その CPU と共有するハードウェアの数は少なくなっていきます。ルート PG は、システム内のすべての CPU を含み、相互に共有するハードウェアがもっとも少ない CPU のグループを表します (PG 階層の例については、後述の「使用例」を参照してください)。

マシンに、パフォーマンス関連のハードウェアの共有関係がない場合、pgstat は、システム内のすべての CPU を含むルート PG のみを表示します。

デフォルトで、pgstat は次の操作を行います。

- 1 秒間の間隔にわたって、PG 階層内の全 PG のハードウェア使用状況およびソフトウェア使用状況を測定します。
- PG の使用状況を深さ優先順で表示し、PG がどのように相互に関連しているかをインデントを使用してわかりやすく示します。
- 一定間隔の時間が経過するごとに、それぞれの PG について、ID、共有関係、ハードウェア負荷、ソフトウェア負荷、およびオンラインの CPU を表示します。

間隔および回数を引数として与えることにより、サンプリング間隔の秒数と、指定した PG の使用状況を測定および表示する回数を指定できます。

オプションを指定することで、出力をさらに調整し、特定の方法で出力を整理し、関心のある PG を指定できます (詳細は、後述の「オプション」を参照)。

指定した PG の使用状況がサポートされていない場合にはハイフン (「-」) が表示され、使用状況を利用できない場合には疑問符 (?) が表示されます。ハードウェア使用状況の測定に CPU ハードウェアパフォーマンスカウンタが必要なシステムでは、attributes(5)、cputrack(1)、dtrace(1M) のほか、libcpc(3LIB) を使用するほかのアプリケーションなど、cpc(3CPC) コンシューマによってカウンタが使用されているために使用状況を利用できない場合があります。

オプション

次のオプションがサポートされています。

`-A`
複数の間隔にわたって `pgstat` を実行している場合に、使用状況データの要約を表示します。

`-c` *processor_id...*
指定した CPU を含む PG に関する使用状況を表示します。CPU は、CPU ID のコンマ区切りのリストとして指定できます。ハイフン (「-」) を使用すると、連続した範囲の CPU ID を指定できます (たとえば `0-3`)。

`-C`
各 PG 内の CPU ごとの使用状況を表示します。

`-h`
簡単なヘルプメッセージを表示し、終了ステータス 0 で終了します。

`-p`
PG に対応する物理的関係を表示します。PG に、システム全体、プロセッサコア、またはチップと同じ CPU がある場合、それぞれ `system`、`core`、または `chip` が、PG の共有関係のあとに角括弧 (「`[]`」) で囲まれて表示されます。

`-P` *pg,...*
指定した PG の使用状況を表示します。複数の PG は、PG ID のコンマ区切りのリストとして指定できます。ハイフン(「-」) を使用すると、連続した範囲の PG ID を指定できます (たとえば `0-3`)。

`-r` *string1, string2,...*
指定された文字列のいずれかに一致した共有関係名を持つ PG に関する使用状況のみを表示します。文字列は、完全な関係名でも、1 つ以上の関係名の一部でも構いません。文字列の照合では大文字と小文字は区別されません。

複数の `-r` オプションを入力できますが、この場合、指定された文字列のいずれかが含まれる関係名を持つすべての PG に一致することになります。

`-R` *string1, string2,...*
指定された名前以外の共有関係名を持つ PG に関する情報のみを表示します。

文字列の照合は、前述の `-r` オプションの説明と同じです。複数の `-R` オプションを入力できます。

`-s` *key*
指定されたキーで、出力行を降順にソートします。指定されたキーは、次のいずれかになります。

`pg`
PG ID の順にソートします。

`hwload`
ハードウェア使用状況の順にソートします。

swload
ソフトウェア使用状況の順にソートします。

user
ユーザー時間の順にソートします。

sys
システム時間の順にソートします。

idle
アイドル時間の順にソートします。

depth
PG ツリーをルートからリーフに下っていく順番で、深さ優先でソートします (デフォルト)。

breadth
PG ツリーをルートからリーフに下っていく順番で、幅優先でソートします。

-S *key*
指定されたキーで、出力行を昇順にソートします。使用可能なキー値は、-s オプションの場合と同じです。

-t *number*
指定された整数値に対応する PG の最上位番号を表示します。

-T u|d
サンプリング間隔ごとのタイムスタンプを、UNIX 時間 (time(2) を参照) で表示するか、date(1) で使用される標準の日付形式で表示します。

-v
ハードウェアの使用状況および容量、ソフトウェアのユーザー時間、システム時間、アイドル時間など、各 PG に関する追加情報を表示します。

出力の見出し

pgstat は、次に示す列の見出しをその意味と一緒に表示します。

PG
プロセッサグループ ID。

RELATIONSHIP
PG の共有関係。

HW
パーセント単位でのハードウェア負荷 (間隔あたりの UTIL/CAP として計算)。

UTIL
PG の共有ハードウェアコンポーネントの、時間間隔あたりのハードウェア使用状況。これは大きな数値になる可能性があるので、K、M、B、および T が、それぞれ 1000、100 万、10 億、1 兆を示すために使用されます。

CAP
: PG の共有ハードウェアコンポーネントの、時間間隔あたりのおおよその最大使用状況見込み。これは大きな数値になる可能性があるので、K、M、B、および T が、それぞれ 1000、100 万、10 億、1 兆を示すために使用されます。

SW
: パーセント単位でのソフトウェア負荷 ((USR + SYS) / (USR + SYS + IDLE) で計算)。

USR
: PG 内の CPU でソフトウェアスレッドがユーザーモードで実行していた時間の、時間間隔に対するパーセント。

SYS
: PG 内の CPU でソフトウェアスレッドがシステムモードで実行していた時間の、時間間隔に対するパーセント。

IDLE
: PG 内の CPU でソフトウェアスレッドが実行していなかった時間の、時間間隔に対するパーセント。

CPUS
: PG 内の CPU の CPU ID。

使用例

次の例では、システムには、8 つのコアと 32 のストランドを備えた 1 つの UltraSPARC T1 プロセッサチップが搭載されています。

例 1 指定した期間での使用状況を表示する

次のコマンドは、直前の 2 秒間にわたるすべての PG の使用状況を表示します。

```
$ pgstat 1 2
PG  RELATIONSHIP            HW    SW  CPUS
 0  System                   -  0.4%  0-31
 3   Data_Pipe_to_memory     -  0.4%  0-31
 2    Floating_Point_Unit   0%  0.4%  0-31
 1     Integer_Pipeline     0%    0%  0-3
 4     Integer_Pipeline     0%    0%  4-7
 5     Integer_Pipeline     0%    0%  8-11
 6     Integer_Pipeline     0%  0.2%  12-15
 7     Integer_Pipeline     0%    0%  16-19
 8     Integer_Pipeline   2.8%  2.7%  20-23
 9     Integer_Pipeline   0.1%  0.2%  24-27
10     Integer_Pipeline     0%    0%  28-31
PG  RELATIONSHIP            HW    SW  CPUS
 0  System                   -  0.4%  0-31
 3   Data_Pipe_to_memory     -  0.4%  0-31
 2    Floating_Point_Unit   0%  0.4%  0-31
 1     Integer_Pipeline     0%  0.2%  0-3
 4     Integer_Pipeline     0%    0%  4-7
 5     Integer_Pipeline     0%    0%  8-11
```

例 1　指定した期間での使用状況を表示する　　*(続き)*

```
 6     Integer_Pipeline     0%    0%  12-15
 7     Integer_Pipeline     0%    0%  16-19
 8     Integer_Pipeline   3.1%  2.5%  20-23
 9     Integer_Pipeline     0%    0%  24-27
10     Integer_Pipeline     0%  0.2%  28-31
```

例 2　整数パイプラインに関する情報を表示する

次のコマンドは、直前の 2 秒間にわたりもっとも頻繁に使用された 2 つの整数パイプラインに関する詳細を表示します。

```
$ pgstat -v -t 2 -r 'Integer_Pipeline' 1 2
PG  RELATIONSHIP         HW UTIL  CAP     SW    USR    SYS   IDLE CPUS
 1  Integer_Pipeline   0.2% 2.2M 1.4B   0.2%   0.0%   0.2%  99.8% 0-3
 4  Integer_Pipeline  13.1% 181M 1.4B  14.9%   0.0%  14.9%  85.1% 4-7
PG  RELATIONSHIP         HW UTIL  CAP     SW    USR    SYS   IDLE CPUS
 1  Integer_Pipeline   0.2% 1.9M 1.2B   0.2%   0.0%   0.2%  99.8% 0-3
 4  Integer_Pipeline  13.1% 163M 1.2B  14.9%   0.0%  14.9%  85.1% 4-7
```

例 3　指定した期間でのコアの使用状況を表示する

次のコマンドは、直前の 2 秒間にわたるコアの使用状況に関する情報を表示します。

```
$ pgstat -A 60 2
PG  RELATIONSHIP              HW      SW  CPUS
 0  System                     -   56.9%  0-31
 3   Data_Pipe_to_memory       -   56.9%  0-31
 2    Floating_Point_Unit   0.0%   56.9%  0-31
 1     Integer_Pipeline    36.7%   58.7%  0-3
 4     Integer_Pipeline    41.9%   58.3%  4-7
 5     Integer_Pipeline    31.0%   58.0%  8-11
 6     Integer_Pipeline    30.7%   57.9%  12-15
 7     Integer_Pipeline    30.1%   55.8%  16-19
 8     Integer_Pipeline    40.2%   54.8%  20-23
 9     Integer_Pipeline    35.0%   56.0%  24-27
10     Integer_Pipeline    40.3%   55.8%  28-31
PG  RELATIONSHIP              HW      SW  CPUS
 0  System                     -   10.7%  0-31
 3   Data_Pipe_to_memory       -   10.7%  0-31
 2    Floating_Point_Unit   0.0%   10.7%  0-31
 1     Integer_Pipeline     9.0%   10.7%  0-3
 4     Integer_Pipeline     9.6%   10.8%  4-7
 5     Integer_Pipeline     8.6%    9.9%  8-11
 6     Integer_Pipeline    10.5%   11.9%  12-15
 7     Integer_Pipeline     9.1%   10.4%  16-19
 8     Integer_Pipeline     9.6%   10.9%  20-23
 9     Integer_Pipeline     8.9%   10.0%  24-27
```

例 3 指定した期間でのコアの使用状況を表示する (続き)

```
10     Integer_Pipeline     9.5%   10.7%  28-31

       SUMMARY: UTILIZATION OVER 120 SECONDS

                              ------HARDWARE------ ------SOFTWARE------
PG  RELATIONSHIP             MIN    AVG    MAX    MIN    AVG    MAX CPUS
 0  System                    -      -      -  10.7%  10.7%  56.9% 0-31
 3   Data_Pipe_to_memory      -      -      -  10.7%  10.7%  56.9% 0-31
 2    Floating_Point_Unit   0.0%   0.0%   0.0%  10.7%  10.7%  56.9% 0-31
 1     Integer_Pipeline     9.0%   8.5%  36.7%  10.7%  10.7%  58.7% 0-3
 4     Integer_Pipeline     9.6%   9.1%  41.9%  10.8%  10.8%  58.3% 4-7
 5     Integer_Pipeline     8.6%   8.1%  31.0%   9.9%   9.9%  58.0% 8-11
 6     Integer_Pipeline    10.5%  10.0%  30.7%  11.9%  11.9%  57.9% 12-15
 7     Integer_Pipeline     9.1%   8.6%  30.1%  10.4%  10.4%  55.8% 16-19
 8     Integer_Pipeline     9.6%   9.1%  40.2%  10.9%  10.9%  54.8% 20-23
 9     Integer_Pipeline     8.9%   8.4%  35.0%  10.0%  10.0%  56.0% 24-27
10     Integer_Pipeline     9.5%   8.9%  40.3%  10.7%  10.7%  55.8% 28-31
```

終了ステータス

次の終了値が返されます。

0
: 正常終了。

1
: システムから PG 情報を取得できません。

2
: 指定された間隔、回数、またはすべての CPU、PG、および共有関係が無効です。

3
: 構文が正しくない。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	不確実

コマンド行オプションおよび出力は「不確実」です。

関連項目

cputrack(1), attributes(5), dtrace(1M), pginfo(1M), cpc(3CPC), libcpc(3LIB), attributes(5)

名前

picld – PICL daemon

形式

```
/usr/lib/picl/picld
```

機能説明

The Platform Information and Control Library (PICL) provides a mechanism to publish platform-specific information for clients to access in a platform-independent way. `picld` maintains and controls access to the PICL information from clients and plug-in modules. The daemon is started in both single-user and multi-user boot mode.

Upon startup, the PICL daemon loads and initializes the plug-in modules. These modules use the `libpicltree(3PICLTREE)` interface to create nodes and properties in the PICL tree to publish platform configuration information. After the plug-in modules are initialized, the daemon opens the PICL daemon door to service client requests to access information in the PICL tree.

PICL Tree

The PICL tree is the repository of all the nodes and properties created by the plug-in modules to represent the platform configuration. Every node in the PICL tree is an instance of a well-defined PICL class. The name of the base PICL class is `picl`, which defines a basic set of properties that all nodes in the tree must possess. Two of those properties are `name` and `_class`, where `name` contains the name of the node, and the `_class` contains the PICL class name of the node. Certain nodes in the PICL tree have well-known names. For example, the name of the root node of the PICL tree is `/` and the name of the root node of the sub-tree containing platform device nodes is `platform`.

PICL plug–in Modules

The PICL plug-in modules are shared objects that publish platform-specific data in the PICL tree. They are located in well-known directories so that the daemon can locate and load them.

Plug-in modules are located in one of the following plug-in directories depending on the plaform-specific nature of the data that they collect and publish:

```
/usr/platform/‘uname -i‘/lib/picl/plugins
/usr/platform/‘uname -m‘/lib/picl/plugins
```

A plug-in module can specify its dependency on another plug-in module using the `-l` or `-R` linker option. The plug-ins are loaded by the daemon using `dlopen(3C)` according to the specified dependencies. Each plug-in module must define a `.init` section, which is executed when the plug-in module is loaded, to register themselves with the daemon. See `picld_plugin_register(3PICLTREE)` for additional information on plug-in registration.

The plug-in modules use the `libpicltree(3PICLTREE)` interface to publish nodes and properties in the PICL tree so that clients can access them.

When the PICL daemon invokes the initialization routine of the plug-in module, the plug-in collects the platform information and creates nodes and/or properties to represent the configuration in the PICL tree. A plug-in can create additional threads to monitor the platform configuration and update the PICL tree with any changes. This enables a PICL plug-in to operate as a daemon within the PICL framework.

An environmental monitor is an example of a plug-in module that uses a thread to monitor the temperatures and fan speeds of the platform, then publishes the environmental information in the PICL tree so clients can access them.

Clients use the libpicl(3PICL) interface to send requests to picld for accessing the PICL tree.

終了ステータス

picld does not return an exit status.

ファイル

/var/run/picld_door　PICL daemon door

/usr/lib/picl/picld　PICL daemon

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/picl

関連項目

svcs(1), svcadm(1M), dlopen(3C), libpicl(3PICL), libpicltree(3PICLTREE), picld_log(3PICLTREE), picld_plugin_register(3PICLTREE), attributes(5), smf(5)

注意事項

The picld service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/picl
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

ping – ICMP (ICMP6) ECHO_REQUEST パケットをネットワーク上のホストに送信する

形式

```
/usr/sbin/ping host [timeout]

/usr/sbin/ping -s [-l | -U] [-abdlLnrRv] [-A addr_family]
     [-c traffic_class] [-g gateway [-g gateway...]]
     [-N next_hop_router] [-F flow_label] [-I interval]
     [-i interface] [-P tos] [-p port] [-t ttl] host
     [data_size] [npackets]
```

機能説明

ping ユーティリティーは、ICMP (IPv6 では ICMP6) プロトコルの ECHO_REQUEST データグラムを利用して、指定されたホスト (*host*) またはネットワークゲートウェイ (*gateway*) から ICMP (ICMP6) ECHO_RESPONSE を引き出します。*host* が応答すると、ping は次のように表示します。

```
host is alive
```

この文字列を標準出力に書き出して終了します。そうでない場合は、タイムアウト (*timeout*) の秒数が経過すると、次の文字列を書き出します。

```
no answer from host
```

timeout のデフォルト値は 20 秒です。

s フラグを指定すると、1 秒ごと (間隔は -I で調整) に 1 つのデータグラムを送信し、ECHO_RESPONSE を受信するたびに 1 行を出力します。応答がないと、ping は出力を生成しません。2 番目の形式では、ping は往復時間とパケットロスの統計値を計算し、終了時またはタイムアウト発生時にこれらの情報の要約を表示します。デフォルトの *data_size* は 56 バイトです。*data_size* コマンド行引数でサイズを指定することもできます。省略可能な *npackets* を指定すると、ping は *npackets* 個の要求を送信するか、*npackets* 個の応答を受信するまで ping 要求を送信し続けます。

ping を障害分離のために使用する場合は、最初にローカルホストに対して ping を実行し、ローカルネットワークインタフェースが動作していることを確認してください。

オプション

次のオプションがサポートされています。

-A *addr_family*

ターゲットホストのアドレスファミリを指定します。 *addr_family* には inet または inet6 を指定できます。アドレスファミリによって使用するプロトコルが決まります。引数が inet の場合は IPv4 を使用します。inet6 の場合は IPv6 を使用します。

デフォルトでは、IP アドレスではなくホストの名前が指定され、ネームサービスのデータベースに有効な IPv6 アドレスが存在する場合、ping はそのアドレスを使用します。それ以外の場合、ネームサービスのデータベースに IPv4 アドレスが含まれていれば、IPv4 アドレスを使用しようとします。

デフォルトの動作を上書きするには、アドレスファミリとして inet または inet6 を指定します。指定した引数が inet の場合、ping はホスト名に関連付けられた IPv4 アドレスを使用します。アドレスが存在しない場合、ping はホストが不明であることを示して終了します。ネームサービスのデータベースに IPv6 アドレスが存在するかどうかは確認されません。

指定した引数が inet6 の場合、ping はホスト名に関連付けられた IPv6 アドレスを使用します。アドレスが存在しない場合、ping はホストが不明であることを示して終了します。

-F *flow_label*
プローブパケットのフローラベルを指定します。0 から 1048575 までの整数を指定する必要があります。このオプションは IPv6 でのみ有効です。

-D
フラグメントを無効にします。IPv4 の場合、これはフラグメント禁止 (DF) ビットを設定することを意味します。IPv4 および IPv6 の場合、これはデータグラムが送信されてもフラグメントを許可しないことを意味します。*data_size* が MTU を超えている場合、ping は、メッセージが長すぎるために送信が失敗したと報告することがあります。

-I *interval*
統計モードを有効にし、連続送信の間隔を指定します。デフォルトは 1 秒です。-s オプションの説明を参照してください。

-L
マルチキャストパケットのループバックを無効にします。通常、メンバーは送信インタフェース上のホストグループに属し、マルチキャストパケットのコピーがローカルマシンに配信されます。

-N *next_hop_router*
次のホップのルーターを指定して、プローブパケットがパスに沿って指定したルーターを経由してターゲットホストに到着するようにします。このオプションを使用すると、基本的にシステムのルーティングテーブルがバイパスされ、プローブパケットのヘッダーが変更されないままになります。指定できる次のホップのルーターは 1 つのみです。

-P *tos*
プローブパケットの Type of Service (*tos*) を指定された値に設定します。デフォルトは 0 です。0 から 255 までの整数を指定する必要があります。同じパス内のゲートウェイは、プローブパケットに設定された *tos* の値に応じてプローブパケットの経路を変更できます。このオプションは IPv4 でのみ有効です。

-R
経路を記録します。IPv4 の経路記録オプションが設定され、IPv4 ヘッダーの内部にパケットの経路が保存されます。経路記録の内容は、-v オプションと -s オプションを指定した場合にのみ出力されます。ターゲットホストが複数のエ

コー間で経路記録オプションを保持する場合や、-l オプションを指定した場合は、返送パケットのみに経路記録オプションが設定されます。このオプションは IPv4 でのみ有効です。

-U
ICMP (ICMP6) パケットの代わりに UDP パケットを送信します。ping は、ターゲットホストから ICMP (ICMP6) PORT_UNREACHABLE が返送されることを期待して、UDP パケットを各ポートに順に送信します。

-a
マルチホームの宛先のすべてのアドレス (IPv4 と IPv6 の両方) に対して ping を実行します。出力は、宛先の各 IP アドレスに対して 1 回ずつ ping を実行した場合と同じように表示されます。このオプションを -A と同時に指定すると、ping は指定されたアドレスファミリに含まれるアドレスだけを検索します。-s オプションとともに使用して *npackets* を指定しないと、ping は宛先アドレスをラウンドロビン方式で継続的に検索します。*npackets* を指定すると、ping は宛先の各 IP アドレスに *npackets* 個のプローブを送信して終了します。

-b
この接続でのみ大域 IPsec ポリシーをバイパスし、暗号化せずにパケットを送受信します。このオプションは、ネットワーク接続のトラブルシューティングを IPsec と切り離して行うために使用できます。このオプションでは、システム全体のポリシーがこの接続でバイパスされるため、使用できるのはスーパーユーザーまたは sys_net_config 特権が付与されたユーザーのみです。

-c *traffic_class*
プローブパケットのトラフィッククラスを指定します。0 から 255 までの整数を指定する必要があります。パス内のゲートウェイは、プローブパケットに設定された *traffic_class* の値に応じてプローブパケットの経路を変更できます。このオプションは IPv6 でのみ有効です。

-d
SO_DEBUG ソケットオプションを設定します。

-g *gateway*
ルーズソースルートゲートウェイを指定して、プローブパケットがパスに沿って指定したホストを経由してターゲットホストに到着するようにします。ゲートウェイの最大数は IPv4 で 8、IPv6 で 127 です。IPv6 のゲートウェイ数は、リンクの MTU などの要因によってさらに制限されることがあります。

-i *interface_address*
IPv4 のマルチキャストパケット、および IPv6 のマルチキャストパケットとユニキャストパケットの両方に使用する送信インタフェースアドレスを指定します。マルチキャストパケットのデフォルトのインタフェースアドレスは、(ユニキャストの) ルーティングテーブルから決定されます。*interface_address* には、リテラル IP アドレス (10.123.100.99 など)、インタフェース名 (eri0 など)、またはインタフェースインデックス (2 など) を指定できます。

`-l`
ルーズソースルーティングを使用して、指定したホストとの間でプローブパケットを送受信します。通常は `-R` オプションとともに使用します。`-g` を使ってゲートウェイを指定すると、それらのゲートウェイを 2 回ずつ (つまり、宛先への送信と宛先からの受信の両方向で) 経由します。`-U` オプションを指定すると、このオプションは無視されます。

`-n`
ネットワークアドレスを数値で表示します。通常、`ping` は受信したパケットから抽出した IP アドレスに対して名前の逆検索を実行します。`-n` オプションを指定すると、逆検索がブロックされるため、`ping` はホスト名ではなく IP アドレスを出力します。

`-p` *port*
プローブで使用されるベース UDP ポート番号を設定します。このオプションは `-U` オプションとともに使用します。デフォルトのベースポート番号は 33434 です。`ping` ユーティリティーは、このベースを起点にして UDP パケットの宛先ポート番号を設定し、プローブごとに 1 ずつ増分します。

`-r`
通常のルーティングテーブルをバイパスし、接続しているネットワーク上のホストに直接送信します。直接接続しているネットワーク上にホストがない場合は、エラーが返されます。このオプションを使用すると、ルーターデーモンによって削除されたインタフェースを介してローカルホストへの `ping` を実行できます。`in.routed(1M)` を参照してください。

`-s`
1 秒に 1 つのデータグラムを送信して、統計情報を収集します。

`-t` *ttl*
ユニキャストパケットおよびマルチキャストパケットの IPv4 生存期間または IPv6 ホップ制限を指定します。ユニキャストパケットのデフォルトの生存期間 (ホップ制限) は、`ipadm(1M)` `set-prop` サブコマンドで `icmp_ipv4_ttl` プロパティー (IPv4) および `icmp_ipv6_hoplimit` プロパティー (IPv6) を使って設定できます。マルチキャストパケットのデフォルトの生存期間 (ホップ制限) は 1 ホップです。「使用例」を参照してください。詳細については、`ipadm(1M)` を参照してください。

注 – プロパティー名が「_」(下線) で始まる場合があります。これらはプロトコルの非公開プロパティーであり、変更または削除される可能性があります。`ipadm(1M)` を参照してください。

`-v`
詳細情報を出力します。ターゲットホストからの応答以外の、すべての ICMP (ICMP6) パケットを表示します。

オペランド

host
ネットワーク上のホスト

使用例

例 1　IPv6 で ping を使用する

この例では、ping を使ってホスト xyz のすべての IPv6 アドレスにプローブパケットを 1 つずつ送信します。ユーザーによる割り込みがあるまで 1 秒間隔で ICMP6 ECHO_REQUEST が送信されます。

```
istanbul% ping -s -A inet6 -a xyz
PING xyz: 56 data bytes
64 bytes from xyz (4::114:a00:20ff:ab3d:83ed): icmp_seq=0. time=0.479 ms
64 bytes from xyz (fec0::114:a00:20ff:ab3d:83ed): icmp_seq=1. time=0.843 ms
64 bytes from xyz (4::114:a00:20ff:ab3d:83ed): icmp_seq=2. time=0.516 ms
64 bytes from xyz (fec0::114:a00:20ff:ab3d:83ed): icmp_seq=3. time=4.943 ms
64 bytes from xyz (4::114:a00:20ff:ab3d:83ed): icmp_seq=4. time=0.485 ms
64 bytes from xyz (fec0::114:a00:20ff:ab3d:83ed): icmp_seq=5. time=2.201 ms
^C
----xyz PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/stddev = 0.479/1.583/4.943/1.823
```

例 2　ipadm を使用してホップ制限を設定する

次のコマンドは、ipadm(1M) を使用して IPv4 および IPv6 ホップ制限を設定します。

```
# ipadm set-prop -p _ipv6_hoplimit=100 icmp
# ipadm set-prop -p _ipv4_ttl=100 icmp
```

終了ステータス

次の終了値が返されます。

0
正常終了。マシンは稼働しています。

0 以外
エラーが発生。不正な形式の引数が指定されたか、マシンが稼働していませんでした。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	network/ping

関連項目

ifconfig(1M), in.routed(1M), ipadm(1M), ndd(1M), netstat(1M), rpcinfo(1M), traceroute(1M), attributes(5), icmp(7P), icmp6(7P)

名前

pkg2du – convert driver packages to Driver Update format

形式

```
/usr/bin/pkg2du [-f] [-v] [-d dir] [-o iso] [-l label]
     [-r release] pkg [pkg ...]
```

機能説明

The `/usr/bin/pkg2du` utility takes one or more packages as input and converts them to Driver Update (DU) format. If the `-d` option is specified, the resulting DU directory tree is placed in the directory *dir*. If the `-o` option is specified, a Solaris ISO image of the DU directory tree is written in the file *iso*. The ISO image can be burned onto CD/DVD using `cdrw(1)` or `cdrecord(1)` (not a SunOS man page) and used during Solaris installation.

At least one of the `-d` and `-o` options must be specified. If both are specified, then both an ISO image and a directory tree are generated.

オプション

The following options are supported:

`-d` *dir*
: Directory where the DU directory should be created.

`-o` *iso*
: Create a Solaris ISO image of the DU directory.

`-f`
: If *dir*`/DU` or *iso* exists, remove it without asking first.

`-l` *label*
: Label/volume name of the ISO image (if `-o` option is specified).

`-r` *release*
: Solaris release number to use. It takes the form of the return from `uname -r` command, for example, `5.10`. If unspecified, the release number of the currently running Solaris is used.

`-v`
: Verbose. Multiple `-v` options increase verbosity.

オペランド

The following operands are supported:

pkg [*pkg*...]
: One or more packages to be converted to DU format.

使用例

例 1　Creating a DU CD/DVD

The following commands create a DU CD or DVD containing packages `SUNWfoo` and `SUNWbar`.

```
# /usr/bin/pkg2du -r 5.10 -o my.iso SUNWfoo SUNWbar
# /usr/bin/cdrw -i my.iso
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

cdrw(1), mkbootmedia(1M), attributes(5)

mkisofs(8) (not a SunOS man page)

名前

pkgadd – transfer software packages to the system

形式

```
pkgadd [-nv] [-a admin] [-G] [-x proxy]
     [ [-M] -R root_path] [-r response] [-k keystore]
     [-P passwd] [-V fs_file]
     [-d device | -d datastream pkginst | all]
     [pkginst | -Y category [, category]...]

pkgadd -s [-d device | -d datastream pkginst | all]
     [pkginst | -Y category [, category]...]
```

機能説明

pkgadd transfers the contents of a software package from the distribution medium or directory to install it onto the system. Used without the -d *device* source specifier, pkgadd looks in the default spool directory (/var/spool/pkg) for the package. Used with the -s option, it writes the package to a spool directory instead of installing it.

The pkgadd utility requires an amount of temporary space the size of the package that is being installed. pkgadd determines which temporary directory to use by checking for the existance of the $TMPDIR environment variable. If $TMPDIR is not defined, pkgadd uses P_tmpdir from stdio.h. P_tmpdir has a default of /var/tmp/.

Certain unbundled and third-party packages are no longer entirely compatible with the latest version of pkgadd. These packages require user interaction throughout the installation and not just at the very beginning, or require that their request scripts be run as the root user.

To install these older packages (released prior to Solaris 2.4), set the following environment variable: NONABI_SCRIPTS=TRUE

As long as this environment variable is set, pkgadd permits keyboard interaction throughout the installation and package request scripts are run as root.

If you have package request scripts that require running as user root (instead of noaccess [the default] or user install), use the rscript_alt parameter in the admin(4) file to make an appropriate selection. See admin(4).

Note that, in Solaris 8 and Solaris 9, the default user when running a request script was either root or nobody, depending on the operating system's patch level. In the current release, the default user is noaccess.

When running pkgadd in the global zone (see zones(5)), a package that contains a request script (see pkgask(1M)) is added only to the global zone. The package is not propagated to any current or yet-to-be-installed non-global zone. This behavior mimics the effect of the -G option, described below.

Package commands are largefile(5)-aware. They handle files larger than 2 GB in the same way they handle smaller files. In their current implementations, pkgadd, pkgtrans(1) and other package commands can process a datastream of up to 4 GB.

The -d, -Y, and *pkginst* arguments shown in the SYNOPSIS are described under OPERANDS, following OPTIONS.

オプション

The supported options are described as follows. The `-d` *device* source specifier is described under OPERANDS, below.

`-a` *admin*
Define an installation administration file, *admin*, to be used in place of the default administration file. The token `none` overrides the use of any *admin* file, and thus forces interaction with the user. Unless a full path name is given, `pkgadd` first looks in the current working directory for the administration file. If the specified administration file is not in the current working directory, `pkgadd` looks in the `/var/sadm/install/admin` directory for the administration file.

`-G`
Add package(s) in the current zone only. When used in the global zone, the package is added to the global zone only and is not propagated to any existing or yet-to-be-created non-global zone. When used in a non-global zone, the package(s) are added to the non-global zone only.

This option causes package installation to fail if, in the `pkginfo` file for a package, `SUNW_PKG_ALLZONES` is set to true. See `pkginfo(4)`.

`-k` *keystore*
Use *keystore* as the location from which to get trusted certificate authority certificates when verifying digital signatures found in packages. If no keystore is specified, then the default keystore locations are searched for valid trusted certificates. See `KEYSTORE LOCATIONS` for more information.

`-M`
Instruct `pkgadd` not to use the `$`*root_path*`/etc/vfstab` file for determining the client's mount points. This option assumes the mount points are correct on the server and it behaves consistently with Solaris 2.5 and earlier releases.

`-n`
Installation occurs in non-interactive mode. Suppress output of the list of installed files. The default mode is interactive.

`-P` *passwd*
Password to use to decrypt keystore specified with `-k`, if required. See `PASS PHRASE ARGUMENTS` for more information about the format of this option's argument.

`-r` *response*
Identify a file or directory which contains output from a previous `pkgask(1M)` session. This file supplies the interaction responses that would be requested by the package in interactive mode. *response* must be a full pathname.

`-R` *root_path*
Define the full path name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*. The *root_path* may be specified when installing to a client from a server (for example, `/export/root/client1`).

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-s` *spool*
Write the package into the directory *spool* instead of installing it.

`-v`
Trace all of the scripts that get executed by `pkgadd`, located in the *pkginst*`/install` directory. This option is used for debugging the procedural and non-procedural scripts.

`-V` *fs_file*
Specify an alternative *fs_file* to map the client's file systems. For example, used in situations where the `$`*root_path*`/etc/vfstab` file is non-existent or unreliable.

`-x` *proxy*
Specify a HTTP[S] proxy to use when downloading packages The format of proxy is *host*:*port*, where *host* is the hostname of the HTTP[S] proxy, and *port* is the port number associated with the proxy. This switch overrides all other methods of specifying a proxy. See ENVIRONMENT VARIABLES for more information on alternate methods of specifying a default proxy.

When executed without options or operands, `pkgadd` uses `/var/spool/pkg` (the default spool directory).

オペランド

The following operands are supported:

Sources

By default, pkgadd looks in the `/var/spool/pkg` directory when searching for instances of a package to install or spool. Optionally, the source for the package instances to be installed or spooled can be specified using:

`-d` *device*
`-d` *datastream pkgname*,... | `all`
Install or copy a package from *device*. *device* can be any of the following:

- A full path name to a directory or the identifiers for tape or removable medium (for example, `/var/tmp`).
- A datastream created by `pkgtrans` (see `pkgtrans(1)`).
- A URL pointing to a datastream created by `pkgtrans`. The supported Universal Resource Identifiers (URIs) are `http:` and `https:`.

The second form of the `-d` specifier, above, indicates the syntax you use when specifying a datastream. In this case you must specify either a comma-separated list of package names or the keyword `all`.

Instances

By default, `pkgadd` searches the specified source, and presents an interactive menu allowing the user to select which package instances found on the source are to be installed. As an alternative, the package instances to be installed can be specified using:

pkginst
The package instance or list of instances to be installed. The token `all` may be used to refer to all packages available on the source medium. The format *pkginst*`.*` can be used to indicate all instances of a package.

The asterisk character (`*`) is a special character to some shells and may need to be escaped. In the C-Shell, the asterisk must be surrounded by single quotes (`'`) or preceded by a backslash (`\`).

`-Y` *category*[,*category*...]
Install packages based on the value of the `CATEGORY` parameter stored in the package's `pkginfo(4)` file. All packages on the source medium whose `CATEGORY` matches one of the specified categories will be selected for installation or spooling.

Keystore Locations

Package such as `pkgadd` use a set of trusted certificates to perform signature validation on any signatures found within the packages. If there are no signatures included in the packages then signature validation is skipped. The certificates can come from a variety of locations. If `-k` *keystore* is specified, and *keystore* is a directory, then *keystore* is assumed to be the base directory of the certificates to be used. If *keystore* is a file, then the file itself is assumed to have all required keys and certificates. When `-k` is not specified, then `/var/sadm/security` is used as the base directory.

Within the specified base directory, the store locations to be searched are different based on the application doing the searching and the type of store being searched for. The following directories are searched in the specified order:

1. <*store_dir*>/<*app_name*>/<*store_type*>
2. <*store_dir*>/<*store_type*>

Where <*store_dir*> is the directory specified by `-k`, <*app_name*> is the name of the application doing the searching, and <*store_type*> is one of `keystore` (for private keys), `certstore` (for untrusted public key certificates), or `truststore` (for trusted certificate authority certificates).

For example, when `pkgadd` is run with `-k /export/certs`, then the following locations are successively searched to find the trust store:

1. /export/certs/pkgadd/truststore
2. /export/certs/truststore

This searching order enables administrators to have a single location for most applications, and special certificate locations for certain applications.

Keystore And Certificate Formats

The packaging utilities, such as `pkgtrans`, require access to a set of keys and certificates in order to sign, and optionally verify, packages.

The keystore files found by following the search pattern specified in `KEYSTORE LOCATIONS` must each be a self-contained PKCS#12-format file.

When signing a package with `pkgtrans`, if a `certstore` has more than one public key certificate, then each public key must have a `friendlyName` attribute in order to be identifiable and selectable with the `-a` option when signing packages. In addition, the public key certificate selected with `-a` and found in the `certstore` must have an associated private key in the keystore.

Several browsers and utilities can be used to export and import certificates and keys into a PKCS#12 keystore. For example, a trusted certificate can be exported from Mozilla, and then imported into a PKCS#12 keystore for use with `pkgadd` with the OpenSSL Toolkit.

Pass Phrase Arguments

`pkgtrans` and `pkgadd` accept password arguments, typically using `-p` to specify the password. These allow the password to be obtained from a variety of sources. Both of these options take a single argument whose format is described below. If no password argument is given and a password is required then the user is prompted to enter one: this will typically be read from the current terminal with echoing turned off.

`pass:`*password*
: The actual password is *password*. Because the password is visible to utilities such as `ps` this form should only be used where security is not important.

`env:`*var*
: Obtain the password from the environment variable *var*. Because the environment of other processes is visible on certain platforms this option should be used with caution.

`file:`*pathname*
: The first line contained within *pathname* is the password. *pathname* need not refer to a regular file: it could, for example, refer to a device or named pipe. For example, to read the password from standard input, use `file:/dev/stdin`.

`console`
: Read the password from `/dev/tty`.

使用例

例 1 Installing a Package from a Solaris DVD

The following example installs a package from a Solaris DVD. You are prompted for the name of the package you want to install.

```
example# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product
```

例 2 Installing a Set of Packages from a Datastream

The example command shown below installs all of the packages in the datastream specified by the `-d` source specifier. Prior to this command, this datastream must have been created with the `pkgtrans(1)` command.

```
example# pkgadd -d /var/tmp/datastream all
```

The keyword `all` specifies that all of the packages found in the designated datastream will be installed.

終了ステータス

`0`
Successful completion

`1`
Fatal error.

`2`
Warning.

`3`
Interruption.

`4`
Administration.

`5`
Administration. Interaction is required. Do not use `pkgadd -n`.

`10`
Reboot after installation of all packages.

`20`
Reboot after installation of this package.

環境変数

`HTTPPROXY`
Specifies an HTTP proxy host. Overrides administration file setting, and `http_proxy` environment variable.

`HTTPPROXYPORT`
Specifies the port to use when contacting the host specified by `HTTPPROXY`. Ignored if `HTTPPROXY` is not set.

`http_proxy`
URL format for specifying proxy host and port. Overrides administration file setting.

ファイル

`/var/sadm/install/logs/`
Location where `pkgadd` logs an instance of software installation.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	package/svr4
Interface Stability	Committed

関連項目

`pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgtrans(1)`, `installf(1M)`, `pkgadm(1M)`, `pkgask(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `removef(1M)`, `admin(4)`, `pkginfo(4)`, `attributes(5)`, `largefile(5)`, `zones(5)`

『Application Packaging Developer's Guide』

http://www.openssl.org

注意事項

When transferring a package to a spool directory, the -r, -n, and -a options cannot be used.

The -r option can be used to indicate a directory name as well as a filename. The directory can contain numerous response files, each sharing the name of the package with which it should be associated. This would be used, for example, when adding multiple interactive packages with one invocation of pkgadd. In this situation, each package would need a response file. If you create response files with the same name as the package (for example, pkinst1 and pkinst2), then name the directory in which these files reside after the -r.

The -n option causes the installation to halt if any interaction is needed to complete it.

If the default *admin* file is too restrictive, the administration file may need to be modified to allow for total non-interaction during a package installation. See admin(4) for details.

If a package stream is specified with -d, and a digital signature is found in that stream, the default behavior is to attempt to validate the certificate and signature found. This behavior can be overridden with admin file settings. See admin(4) for more information.

名前

pkgadm – manage packaging and patching system

形式

```
pkgadm addcert [-ty] [-a app] [-k keystore] [-e keyfile]
     [-f format] [-n name] [-P passarg]
     [-p import_passarg] [-R rootpath] certfile

pkgadm removecert [-a app] [-k keystore] -n name
     [-P passarg] [-R rootpath]

pkgadm listcert [-a app] [-f format] [-k keystore] -n name
     [-P passarg] [-o outfile] [-R rootpath]

pkgadm dbstatus [-R rootpath]

pkgadm sync [-R rootpath] [-q]

pkgadm -V

pkgadm -?
```

機能説明

The `pkgadm` utility is used for managing the packaging and patching system. It has several subcommands that perform various operations relating to packaging. The `pkgadm` command includes subcommands for managing certificates and keys used.

Managing Keys and Certificates

`pkgadm` maintains the packaging-system-wide keystore in `/var/sadm/security`, and individual user's certificates in `~/.pkg/security`. The following subcommands operate on the package keystore database:

`addcert`
: Add (import) a certificate into the database, with optional trust. Once added, trusted certificates can be used to verify signed packages and patches. Non-trusted user certificates and their associated keys can be used to sign packages and patches. Added user certificates are *not* used to build certificate chains during certificate verification.

`removecert`
: Removes a user certificate/private key pair, or a trusted certificate authority certificate from the keystore. Once removed, the certificate and keys cannot be used.

`listcert`
: Print details of one or more certificates in the keystore.

`sync`
: Writes the contents file and rolls the contents log file. With use of the `-q` option, forces the contents file server to quit.

Internal Install Database

The Solaris operating system relies upon enhanced System V revision 4 (SVr4) packages as the basis for its software installation and revision management. The package maintenance software stores information about installed packages in an internal database. The `pkgadm` subcomand `dbstatus` is used to determine how the package internal database is implemented. The `dbstatus` command returns a string that indicates the type of internal database in use. In

the current implementation, the `dbstatus` command always returns the string `text`, which indicates that the `contents(4)` package database is inuse. Future releases of Solaris might supply alternative database implementations.

オプション

The following options are supported:

`-a` *app*
: If this option is used, then the command only affects the keystore associated with a particular application. Otherwise, the global keystore is affected.

`-e` *keyfile*
: When adding a non-trusted certificate/key combination, this option can be used to specify the file that contains the private key. If this option is not used, the private key must be in the same file as the certificate being added.

`-f` *format*
: When adding certificates, this specifies the format to expect certificates and private keys in. Possible values when adding are:

  `pem`
  : Certificate and any private key uses PEM encoding.

  `der`
  : Certificate and any private key uses DER encoding.

  When printing certificates, this specifies the output format used when printing. Acceptable values for format are:

  `pem`
  : Output each certificate using PEM encoding.

  `der`
  : Output each certificate using DER encoding.

  `text`
  : Output each certificate in human-readable format.

`-k` *keystore*
: Overrides the default location used when accessing the keystore.

`-n` *name*
: Identifies the entity in the store on which you want to operate. When adding a user certificate, or removing certificates, this name is required. The name is associated with the certificate/key combination, and when adding, can be used later to reference the entity. When printing certificates, if no alias is supplied, then all keystore entities are printed.

`-o` *outfile*
: Output the result of the command to *outfile*. Only used when examining (printing) certificates from the key store. Standard out is the default.

`-P` *passarg*
Password retrieval method to use to decrypt keystore specified with `-k`, if required. See `PASS PHRASE ARGUMENTS` in `pkgadd(1M)` for more information about the format of this option's argument. `console` is the default.

`-p` *import_passarg*
This option's argument is identical to `-P`, but is used for supplying the password used to decrypt the certificate and/or private key being added. `console` is the default.

`-q`
(Applies to `sync` subcommand.) Shuts down the contents file cache daemon.

`-R` *rootpath*
Defines the full name of a directory to use as the root (`/`) path. The default user location of the certificate operations is `${HOME}/.pkg`. If the `-R` option is supplied, the certificates and keys will be stored under <*altroot*>`/var/sadm/security`. Note that this operation fails if the user does not have sufficient permissions to access this directory. The `listcert` command requires read permission, while `addcert` and `removecert` require both read and write permission.

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-t`
Indicates the certificate being added is a trusted CA certificate. The details of the certificate (including the Subject Name, Validity Dates, and Fingerprints) are printed and the user is asked to verify the data. This verification step can be skipped with `-y`. When importing a trusted certificate, a private key should not be supplied, and will be rejected if supplied. Once a certificate is trusted, it can be used as a trust anchor when verifying future untrusted certificates.

`-V`
Print version associated with packaging tools.

`-y`
When adding a trusted certificate, the details of the certificate (Subject name, Issuer name, Validity dates, Fingerprints) are shown to the user and the user is asked to verify the correctness before proceeding. With `-y`, this additional verification step is skipped.

`-?`
Print help message.

オペランド

The following operand is supported:

`certfile`
File containing the certificate and optional private key, used when adding a trust anchor or certificate/key combination. Certificates must be encoded using PEM or binary DER.

Keystore Aliases

All keystore entries (user cert/key and trusted certificate entries) are accessed via unique aliases. Aliases are case-sensitive.

An alias is specified when you add an entity to a keystore using the `addcert` or `trustcert` subcommand. If an alias is not supplied for a trust anchor, the trust anchor's Common Name is used as the alias. An alias is required when adding a signing certificate or chain certificate. Subsequent `pkgcert` or other package tool commands must use this same alias to refer to the entity.

Keystore Passwords

See the `pkgadd(1M)` man page for a description of the passwords supplied to the `pkgadm` utility.

使用例

例 1 Adding a Trust Anchor

The following example adds a well-known and trusted certificate to be used when verifying signatures on packages.

```
example% pkgadm addcert -t /tmp/certfile.pem
```

例 2 Adding a Signing Certificate

The following example adds a signing certificate and associated private key, each of which is in a separate file, which can then be used to sign packages.

```
example% pkgadm addcert -a pkgtrans -e /tmp/keyfile.pem \
/tmp/certfile.pem
```

例 3 Printing Certificates

The following example prints all certificates in the root keystore.

```
example% pkgadm listcert
```

終了ステータス

`0`
: successful completion

`non-zero`
: fatal error

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	package/svr4
Interface Stability	Committed

関連項目

`pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgtrans(1)`, `svcs(1)`, `installf(1M)`, `pkgadd(1M)`, `pkgask(1M)`, `pkgrm(1M)`, `removef(1M)`, `svcadm(1M)`, `admin(4)`, `contents(4)`, `exec_attr(4)`, `pkginfo(4)`, `attributes(5)`, `rbac(5)`, `smf(5)`

『Application Packaging Developer's Guide』

注意事項

The service for pkgadm is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/pkgserv
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

pkgask – stores answers to a request script

形式

`pkgask [-d device] [-R root_path] -r response pkginst...`

機能説明

`pkgask` allows the administrator to store answers to an interactive package (one with a `request` script, that is, a user-created file that must be named `request`). Invoking this command generates a response file that is then used as input at installation time. The use of this response file prevents any interaction from occurring during installation since the file already contains all of the information the package needs.

オプション

The following options are supported

`-d` *device*
: Run the request script for a package on *device*. *device* can be a directory pathname or the identifiers for tape or removable medium (for example, `/var/tmp` and `/dev/dsk/c1d0s0`). The default device is the installation spool directory.

`-R` *root_path*
: Define the full path name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*.

  注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-r` *response*
: Identify a file or directory which should be created to contain the responses to interaction with the package. The name must be a full pathname. The file, or directory of files, can later be used as input to the `pkgadd(1M)` command.

オペランド

The following operands are supported:

pkginst
: Specify the package instance, or list of instances for which `request` scripts will be created. The token `all` may be used to refer to all packages available on the source medium.

終了ステータス

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgtrans(1)`, `installf(1M)`, `pkgadd(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `removef(1M)`, `admin(4)`,`attributes(5)`

『Application Packaging Developer's Guide』

注意事項

The -r option can be used to indicate a directory name as well as a filename. The directory name is used to create numerous response files, each sharing the name of the package with which it should be associated. This would be used, for example, when you will be adding multiple interactive packages with one invocation of pkgadd(1M). Each package would need a response file. To create multiple response files with the same name as the package instance, name the directory in which the files should be created and supply multiple instance names with the pkgask command. When installing the packages, you will be able to identify this directory to the pkgadd(1M) command.

If the default admin file is too restrictive, the administration file may need to be modified to allow for total non-interaction during a package installation. Seeadmin(4) for details.

名前

pkgchk – check package installation accuracy

形式

```
pkgchk [-l | -acfnqvx] [-i file | -]
     [-p path... | -P partial-path...] [-R root_path]
     [ [-m pkgmap [-e envfile]] | pkginst... | -Y category,category...]

pkgchk -d device [-l | -fv] [-i file | -] [-M] [-p path]...
     [-V fs_file]
     [pkginst... | -Y category[,category...]]
```

機能説明

`pkgchk` checks the accuracy of installed files or, by using the `-l` option, displays information about package files. `pkgchk` checks the integrity of directory structures and files. Discrepancies are written to standard error along with a detailed explanation of the problem.

The first synopsis defined above is used to list or check the contents and/or attributes of objects that are currently installed on the system, or in the indicated `pkgmap`. Package names may be listed on the command line, or by default, the entire contents of a machine will be checked.

The second synopsis is used to list or check the contents of a package which has been spooled on the specified device, but not installed. Note that attributes cannot be checked for spooled packages.

オプション

The following options are supported:

`-a`
: Audit the file attributes only and do not check file contents. Default is to check both.

`-c`
: Audit the file contents only and do not check file attributes. Default is to check both.

`-d` *device*
: Specify the device on which a spooled package resides. *device* can be a directory path name or the identifiers for tape or removable medium (for example, `/var/tmp`.

`-e` *envfile*
: Request that the package information file named as *envfile* be used to resolve parameters noted in the specified `pkgmap` file.

`-f`
: Correct file attributes if possible. If used with the `-x` option, this option removes hidden files. When `pkgchk` is invoked with this option, it creates directories, named pipes, links, and special devices if they do not already exist. If the `-d` option calls out an uninstalled package, the `-f` option will only take effect if the package is in directory (not stream) format. All file attributes will be set to agree with the entries in the `pkgmap` file except that setuid, setgid, and sticky bits will not be set in the mode.

`-i` *file* | –
Read a list of path names from *file* or from stdin (–) and compare this list against the installation software database or the indicated `pkgmap` file. Path names that are not contained in *file* or stdin are not checked.

`-l`
List information on the selected files that make up a package. This option is not compatible with the `-a`, `-c`, `-f`, `-g`, and `-v` options.

`-m pkgmap`
Check the package against the package map file, `pkgmap`.

`-M`
Instruct `pkgchk` not to use the `$`*root_path*`/etc/vfstab` file for determining the client's mount points. This option assumes the mount points are correct on the server and it behaves consistently with Solaris 2.5 and earlier releases.

`-n`
Do not check volatile or editable files' contents. This should be used for most post-installation checking.

`-p` *path*
Check the accuracy only of the path name or path names listed. *path* can be one or more path names separated by commas (or by whitespace, if the list is quoted).

To specify a *path* that includes a comma, you must use the `-i` option, described above. See EXAMPLES.

`-P` *partial-path*
Check the accuracy of only the partial path name or path names listed. *partial-path* can be one or more partial path names separated by commas (or by whitespace, if the list is quoted). This option can be used instead of `-p` and is not compatible with the other option. This option matches any path name that contains the string contained in the partial path. See the note about paths that contain commas in the description of `-p`.

`-q`
Quiet mode. Do not give messages about missing files.

`-R` *root_path*
Define the full name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*. The *root_path* may be specified when installing to a client from a server (for example, `/export/root/client1`).

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-v`
Verbose mode. Files are listed as processed.

-V *fs_file*
Specify an alternative *fs_file* to map the client's file systems. For example, used in situations where the $*root_path*/etc/vfstab file is non-existent or unreliable.

-x
Search exclusive directories, looking for files which exist that are not in the installation software database or the indicated pkgmap file.

-Y *category*
Check packages based on the value of the CATEGORY parameter stored in the installed or spooled package's pkginfo(4) file.

オペランド

pkginst
The package instance or instances to be checked. The format *pkginst*.* can be used to check all instances of a package. The default is to display all information about all installed packages.

The asterisk character (*) is a special character to some shells and may need to be escaped. In the C-Shell, an asterisk must be surrounded by single quotes ('') or preceded by a backslash (\);

partial-path
A portion of a path, such as a file or directory name.

使用例

例 1 Using pkgchk for Displaying Package Installation Information

The following example displays package installation information for /usr/bin/ls:

```
example% pkgchk -l -p /usr/bin/ls
```

例 2 Checking on Java Font Properties

The following example displays package installation information for all Java font properties installed on the system.

```
example% pkgchk -l -P font.properties
```

例 3 Specifying a Path That Contains a Comma

Assume you want to specify the path:

```
/platform/SUNW,Netra-T12/lib
```

List this path in a file. Here is one way in which you can do that:

```
example% echo "/platform/SUNW,Netra-T12/lib" > /tmp/p
```

You can then enter:

```
example% pkgchk -i /tmp/p -l
Pathname: /platform/SUNW,Netra-T12/lib
Type: directory
```

例 3 Specifying a Path That Contains a Comma (続き)

```
Expected mode: 0755
Expected owner: root
Expected group: bin
Referenced by the following packages:
        system/core-osar
Current status: installed
```

終了ステータス

`0`
Successful completion.

`>0`
An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`pkginfo(1)`, `pkgtrans(1)`, `pkgadd(1M)`, `pkgask(1M)`, `pkgrm(1M)`, `pkginfo(4)`, `attributes(5)`, `largefile(5)`

『Application Packaging Developer's Guide』

注意事項

Package commands are `largefile(5)`-aware. They handle files larger than 2 GB in the same way they handle smaller files. In their current implementations, `pkgadd(1M)`, `pkgtrans(1)` and other package commands can process a datastream of up to 4 GB.

名前

pkgcond – determine type and capability of target

形式

`/usr/bin/pkgcond [-nv]` *condition*

機能説明

The `pkgcond` command allows you to determine the type of target being operated on (global zone, non-global zone, diskless client, and so forth) and the capabilities available for that type of client (can add a driver, path is writable, and so forth). The `pkgcond` command is intended to be invoked from package and patch scripts, but can also be used in situations that mimic the context of these scripts. See `NOTES` for further guidance.

`pkgcond` has one mandatory argument, a condition. The command tests whether the condition is true for the specified path. The condition can be one of the following:

- `can_add_driver` [*path*]
- `can_remove_driver` [*path*]
- `can_update_driver` [*path*]
- `is_alternative_root` [*path*]
- `is_boot_environment` [*path*]
- `is_diskless_client` [*path*]
- `is_global_zone` [*path*]
- `is_mounted_miniroot` [*path*]
- `is_netinstall_image` [*path*]
- `is_nonglobal_zone` [*path*]
- `is_path_writable` *path*
- `is_running_system` [*path*]
- `is_sparse_root_nonglobal_zone` [*path*]
- `is_what` [*path*]
- `is_whole_root_nonglobal_zone` [*path*]

The *path* argument usually denotes the root of the global zone or non-global zone, or alternate root. If path is optional and not specified, the default is `/`.

The behavior of the `is_what` condition is somewhat special, because it displays results of all other conditions to standard output.

オプション

The following options are supported:

`-n`
: Negate return status (`0` becomes `1` and `1` becomes `0`). It negates results in the case of `is_what` condition.

`-v`
: Verbose mode. Displays detailed data about intermediate checks performed.

使用例

例 1　Listing All Available Information

The following command lists all available information about the current running system, in a user-friendly way.

```
example# pkgcond -n is_what
```

例 2 Determining if Target is an Alternate Root

The following command determines whether an alternate boot environment exists under `/altroot_mount`.

```
example# pkgcond is_alternative_root /altroot_mount
```

終了ステータス

`0`
: Condition is true unless `-n` was specified.

`1`
: Condition is false unless `-n` was specified.

`2`
: Command line usage error.

`3`
: Command failed to perform the test due to a fatal error.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

関連項目

`pkgtrans(1)`, `pkgadd(1M)`, `pkgask(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `pkginfo(4)`, `attributes(5)`

注意事項

Supported usage of `pkgcond` is subject to the following constraints:

1. Do not use `pkgcond` outside of the Solaris marketing release in which it is provided (for example, do not use Solaris 10 `pkgcond` against a Solaris 9 target).
2. Restrict use of the optional path argument according to the following rules:
   - The command `pkgcond` *condition* `$ROOTDIR` can be used in patch level scripts.
   - The command `pkgcond` *condition* `$PKG_INSTALL_ROOT` can be used in package level scripts.
   - A command of the form `pkgcond` *condition without* the optional path argument can be used in any context.

Use of `pkgcond` with an arbitrary path argument is *not* recommended or supported, as the results returned might not be accurate.

名前

pkgrm – remove a package from the system

形式

```
pkgrm [-nv] [-a admin] [ [-A | -M] -R root_path]
     [-V fs_file]
     [pkginst... | -Y category[,category...]]

pkgrm -s spool
     [pkginst... | -Y category[,category...]]
```

機能説明

`pkgrm` will remove a previously installed or partially installed package from the system. A check is made to determine if any other packages depend on the one being removed. If a dependency exists, the action taken is defined in the `admin` file.

The default state for the command is in interactive mode, meaning that prompt messages are given during processing to allow the administrator to confirm the actions being taken. Non-interactive mode can be requested with the `-n` option.

The `-s` option can be used to specify the directory from which spooled packages should be removed.

Certain unbundled and third-party packages are no longer entirely compatible with the latest version of `pkgrm`. These packages require user interaction throughout the removal and not just at the very beginning.

To remove these older packages (released prior to Solaris 2.4), set the following environment variable:`NONABI_SCRIPTS=TRUE` `pkgrm` permits keyboard interaction throughout the removal as long as this environment variable is set.

オプション

The following options are supported:

`-a` *admin*
: Use the installation administration file, `admin`, in place of the default `admin` file. `pkgrm` first looks in the current working directory for the administration file. If the specified administration file is not in the current working directory, `pkgrm` looks in the `/var/sadm/install/admin` directory for the administration file.

`-A`
: Remove the package files from the client's file system, absolutely. If a file is shared with other packages, the default behavior is to not remove the file from the client's file system.

`-M`
: Instruct `pkgrm` not to use the `$`*root_path*`/etc/vfstab` file for determining the client's mount points. This option assumes the mount points are correct on the server and it behaves consistently with Solaris 2.5 and earlier releases.

`-n`
: Non-interactive mode. If there is a need for interaction, the command will exit.

Use of this option requires that at least one package instance be named upon invocation of the command. Certain conditions must exist for a package to be removed non-interactively or a non-restrictive `admin` file needs to be used.

`-R` *root_path*
Defines the full path name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*.

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-s` *spool*
Remove the specified package(s) from the directory *spool*. The default directory for spooled packages is `/var/sadm/pkg`.

`-v`
Trace all of the scripts that get executed by `pkgrm`, located in the *pkginst*`/install` directory. This option is used for debugging the procedural and non-procedural scripts.

`-V` *fs_file*
Specify an alternative *fs_file* to map the client's file systems. Used in situations where the $*root_path*`/etc/vfstab` file is non-existent or unreliable.

`-Y` *category*
Remove packages based on the value of the `CATEGORY` parameter stored in the installed or spooled package's `pkginfo(4)` file. No package with the `CATEGORY` value of `system` can removed from the file system with this option.

オペランド

The following operand is supported:

pkginst
Specifies the package to be removed. The format *pkginst*`.*` can be used to remove all instances of a package.

The asterisk character (`*`) is a special character to some shells and may need to be escaped. In the C-Shell, "`*`" must be surrounded by single quotes (`'`) or preceded by a backslash (\).

使用例

例 1 Removing All Instances of SUNWjunk from client1

The following example removes all instances of `SUNWjunk` from `client1`:

```
example% pkgrm -R /export/root/client1 SUNWjunk*
```

Note the caveat on the use of the `-R` option in the description of that option, above.

終了ステータス

The following exit values are returned:

`0`
Successful completion.

1
Fatal error.

2
Warning.

3
Interruption.

4
Administration.

10
Reboot after removal of all packages.

20
Reboot after removal of this package.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

pkginfo(1), pkgmk(1), pkgparam(1), pkgproto(1), pkgtrans(1), installf(1M), pkgadd(1M), pkgask(1M), pkgchk(1M), removef(1M), admin(4), pkginfo(4), attributes(5), largefile(5)

『Application Packaging Developer's Guide』

注意事項

Package commands are largefile(5)-aware. They handle files larger than 2 GB in the same way they handle smaller files. In their current implementations, pkgadd(1M), pkgtrans(1) and other package commands can process a datastream of up to 4 GB.

名前

plockstat – report user-level lock statistics

形式

```
plockstat [-vACHV] [-n count] [-s depth] [-e secs]
     [-x arg [=val]] command [arg]...

plockstat [-vACHV] [-n count] [-s depth] [-e secs]
     [-x arg [=val]] -p pid
```

機能説明

The `plockstat` utility gathers and displays user-level locking statistics. By default, `plockstat` monitors all lock contention events, gathers frequency and timing data about those events, and displays the data in decreasing frequency order, so that the most common events appear first.

`plockstat` gathers data until the specified command completes or the process specified with the `-p` option completes.

`plockstat` relies on DTrace to instrument a running process or a command it invokes to trace events of interest. This imposes a small but measurable performance overhead on the processes being observed. Users must have the `dtrace_proc` privilege and have permission to observe a particular process with `plockstat`. Refer to the 『Solaris Dynamic Tracing Guide』 for more information about DTrace security features.

オプション

The following options are supported:

`-A`	Watch all lock events. This option is equivalent to `-CH`.
`-C`	Watch contention events.
`-H`	Watch hold events.
`-e` *secs*	Exit after the number of seconds specified have elapsed.
`-n` *count*	Display only the specified number of entries for each output category.
`-s` *depth*	Record a stack trace rather than just the calling function.
`-p` *pid*	Specify a process ID from which `plockstat` is to gather data.
`-v`	Print out a message to indicate that tracing has started.
`-x` *arg*[=*val*]	Enable or modify a DTrace runtime option or D compiler option. The list of options is found in the 『Solaris Dynamic Tracing Guide』 . Boolean options are enabled by specifying their name. Options with values are set by separating the option name and value with an equals sign (=).
`-V`	Print the Dtrace commands used to gather the data. The output can then be used directly with the `dtrace(1M)` command.

オペランド

The following operands are supported:

arg	A string to be passed as an argument to *command*.
command	The name of a utility to be invoked.

count A positive integer value.

pid A process identifier for a process to be monitored.

secs Duration specified as a positive integer number of seconds.

Display Headers

The following headers appear over columns of data in `plockstat` output.

`Count`
: Number of times an event occurred.

`nsec`
: Average duration of an event, in nanoseconds.

`Lock`
: Address of a lock, displayed symbolically if possible.

`Caller`
: Address of a caller, displayed symbolically if possible.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/dtrace
Interface Stability	See below.

The command-line syntax is Committed. The human-readable output is Uncommitted.

関連項目

`dtrace(1M)`, `lockstat(1M)`, `mutex_init(3C)`, `pthread_mutex_lock(3C)`, `pthread_rwlock_rdlock(3C)`, `pthread_rwlock_wrlock(3C)`, `pthread_rwlock_unlock(3C)`, `rwlock(3C)`, `attributes(5)`, `fasttrap(7D)`

『Solaris Dynamic Tracing Guide』

名前

pntadm – DHCP network table management utility

形式

```
pntadm -C [-r resource] [-p path] [-u uninterpreted] network

pntadm -A name_IP_address [-c comment] [-e mm/dd/yyyy]
     [-f num | keywords] [-h client_hostname]
     [-i [-a] client_ID] [-m [-y] macro] [-s server]
     [-r resource] [-p path] [-u uninterpreted] network

pntadm -M name_IP_address [-c comment] [-e mm/dd/yyyy]
     [-f num | keywords] [-h client_hostname]
     [-i [-a] client ID] [-m [-y] macro]
     [-n new_client_IP_address] [-s server] [-r resource]
     [-p path] [-u uninterpreted] network

pntadm -D name_IP_address [-y] [-r resource] [-p path]
     [-u uninterpreted] network

pntadm -P [-v] [-x] [-r resource] [-p path]
     [-u uninterpreted] network

pntadm -R [-r resource] [-p path] [-u uninterpreted] network

pntadm -L [-r resource] [-p path] [-u uninterpreted]

pntadm -B [-v] [batchfile]
```

機能説明

The `pntadm` command is used to manage the Dynamic Host Configuration Protocol (DHCP) network tables. It is used to add and remove networks under DHCP management, and add, delete, or modify IP address records within network tables, or to view tables. For a description of the format of DHCP network tables, see `dhcp_network(4)`.

`pntadm` can be run as root or by other users assigned to the DHCP Management profile. See `rbac(5)` and `user_attr(4)`.

If the networks you want to add are subnetted, you need to update the `netmasks(4)` table.

One of the following options (function flags) must be specified with the `pntadm` command: `-A`, `-B`, `-C`, `-D`, `-L`, `-M`, `-P`, or `-R`.

The `pntadm` utility is obsolete and is subject to removal in a future release of Oracle Solaris.

オプション

The following options are supported:

`-A` *name_IP_address*

Add a client entry with hostname or client IP address, *name_IP_address*, to the named DHCP network table.

The following sub-options are optional:

`-c` *comment*

Comment text. The default is `NULL`.

`-e` *mm/dd/yyyy*
Absolute lease. The default is `0`.

`-f` *num* | *keywords*
Flag value. The default is `00`.

The flag (`-f`) option can be specified either as a single number denoting the intended flag value, or as a series of the following keywords, combined using the plus (+) symbol:

`DYNAMIC` or `00`
Server manager's assignment.

`PERMANENT` or `01`
Lease on entry is permanent.

`MANUAL` or `02`
Administrator managed assignment.

`UNUSABLE` or `04`
Entry is not valid.

`BOOTP` or `08`
Entry reserved for `BOOTP` clients.

For a more detailed description of the flag values, see `dhcp_network(4)`.

`-h` *client_hostname*
Client hostname. The default is NULL.

When the `-h` option is used in this mode, the *client_hostname* is added to the hosts table within the resource used for storing host names (files or DNS). The command will fail if this *client_hostname* is already present in the hosts table.

`-i` *client_ID* [`-a`]
Client identifier [`-a`]. The default is `00`.

The `-i` option modified with `-a` specifies that the client identifier is in ASCII format, and thus needs to be converted to hexadecimal format before insertion into the table.

`-m` *macro* [`-y`]
Macro name. Default is UNKNOWN.

The `-m` option modified with `-y` verifies the existence of the named macro in the `dhcptab` table before adding the entry.

`-s` *server*
Server IP or name. Default is system name (`uname -n`).

`-B`
Activate batch mode. `pntadm` will read from the specified file or from standard input a series of `pntadm` commands and execute them within the same process. Processing many

pntadm commands using this method is much faster than running an executable batchfile itself. Batch mode is recommended for using pntadm in scripts.

The following sub-option is optional:

-v
Display commands to standard output as they are processed.

-C
Create the DHCP network table for the network specified by *network*. See オペランド. For details, see dhcp_network(4) and networks(4).

-D *name_IP_address*
Delete the specified client entry with hostname or client IP address, *name_IP_address*, in the named DHCP network table. (See dhcp_network(4).)

The following sub-option is optional:

-y
Remove associated host table entry. The -y option requests that all hostnames associated with the IP address in the hosts table in the resource be removed.

-L
List the DHCP network tables presently configured, one per line, on standard output. If none are found, no output is printed and an exit status of 0 is returned.

-M *name_IP_address*
Modify the specified client entry with hostname or client IP address, *name_IP_address*, in the named DHCP network table. See dhcp_network(4). The default for the sub-options is what they currently are set to.

The following sub-options are optional.

-c *comment*
New comment text.

-e *mm/dd/yy*
New absolute lease expiration date. Time defaults to 12:00 AM of the day specified.

-f *num* | *keyboard*
New flag value, see explanation following the description of the -A option.

-h *host_name*
New client hostname.

The -h option allows you to change the current *hostname* associated with the IP address or to add a new *hostname* to the hosts table if an entry associated with this IP address does not exist.

-i *client_ID*
New client identifier [-a].

-m *macro* [-y]
Macro name defined in dhcptab.

-n *new_client_IP_address*
New IP address.

-s *server*
New server IP or name.

For more detailed description of the sub-options and flag values, see dhcp_network(4).

-P
Display the named DHCP network table.

The following sub-options are optional:

-v
Display lease time in full verbose format and resolve IP addresses for the clients and server to hostnames.

-x
Display lease time in raw format.

These flag codes are used with the -P sub-options:

-v	-x	Description
D	00	DYNAMIC
P	01	PERMANENT
M	02	MANUAL
U	04	UNUSABLE
B	08	BOOTP

See dhcp_network(4) for information on these sub-options and associated flag codes.

-p *path*
Override the dhcpsvc.conf(4) configuration value for data store resource path, *path* See dhcpsvc.conf(4)

-R
Remove the named DHCP network table. See dhcp_network(4).

-r *data_store_resource*
Override the /etc/inet/dhcpsvc.conf configuration value for RESOURCE= with the *data_store_resource* specified. See the dhcpsvc.conf(4) man page for more details on resource type, and the 『Oracle Solaris DHCP Service Developer's Guide』 for more information about adding support for other data stores.

-u uninterpreted
: Data which will be ignored by pntadm, but passed to the currently configured public module to be interpreted by the data store. This might be used for a database account name or other authentication or authorization parameters required by a particular data store.

オペランド

The following operand is supported:

network
: The network address or network name which corresponds to the dhcp network table. See dhcp_network(4).

使用例

例 1 Creating a Table for the 10.0.0.0 DHCP Network

The following command creates a table for the 10.0.0.0 (subnetted to class C) DHCP network table. Note that if you have an alias for this network in your networks(4) table, you can use that value rather than the dotted Internet Address notation.

```
example# pntadm -C 10.0.0.0
```

例 2 Adding an Entry to the 10.0.0.0 Table

The following command adds an entry to the 10.0.0.0 table in the files resource in the /var/mydhcp directory:

```
example# pntadm -r SUNWfiles -p /var/mydhcp -A 10.0.0.1 10.0.0.0
```

例 3 Modifying the 10.0.0.1 Entry of the 10.0.0.0 Table

The following command modifies the 10.0.0.1 entry of the 10.0.0.0 table, changing the macro name to Green, setting the flags field to MANUAL and PERMANENT:

```
example# pntadm -M 10.0.0.1 -m Green -f 'PERMANENT+MANUAL' 10.0.0.0
```

例 4 Changing the 10.0.0.1 Entry to 10.0.0.2

The following command changes the 10.0.0.1 entry to 10.0.0.2, making an entry in the hosts(4) table called myclient:

```
example# pntadm -M 10.0.0.1 -n 10.0.0.2 -h myclient 10.0.0.0
```

例 5 Setting the Client ID as ASCII

The following command sets the client ID as ASCII aruba.foo.com for the myclient entry:

```
example# pntadm -M myclient -i 'aruba.foo.com' -a 10.0.0.0
```

例 6 Deleting the myclientEntry from the 10.0.0.0 Table

The following command deletes the myclient (10.0.0.2) entry from the 10.0.0.0 table:

```
example# pntadm -D myclient 10.0.0.0
```

例 7 Listing the Configured DHCP Network Tables

The following command lists the configured DHCP network tables:

```
example# pntadm -L
192.168.0.0
10.0.0.0
```

例 8 Executing pntadm Commands in Batch Mode

The following command runs a series of `pntadm` commands contained in a batch file:

```
example# pntadm -B addclients
```

終了ステータス

`0`
Successful completion.

`1`
Object already exists.

`2`
Object does not exist.

`3`
Non-critical error.

`4`
Critical error.

ファイル

- `/etc/inet/dhcpsvc.conf`
- `/etc/inet/hosts`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/dhcp
Interface Stability	Obsolete

関連項目

`dhcpconfig(1M)`, `dhcpmgr(1M)`, `dhcp_network(4)`, `dhcpsvc.conf(4)`, `dhcptab(4)`, `hosts(4)`, `netmasks(4)`, `networks(4)`, `user_attr(4)`, `attributes(5)`, `dhcp(5)`, `dhcp_modules(5)`, `rbac(5)`

『Oracle Solaris DHCP Service Developer's Guide』

『Oracle Solaris の管理: IP サービス』

Alexander, S., and R. Droms, DHCP Options and BOOTP Vendor Extensions, RFC 1533, Lachman Technology, Inc., Bucknell University, October 1993.

Droms, R., Interoperation Between DHCP and BOOTP, RFC 1534, Bucknell University, October 1993.

Droms, R., Dynamic Host Configuration Protocol, RFC 1541, Bucknell University, October 1993.

Wimer, W., Clarifications and Extensions for the Bootstrap Protocol, RFC 1542, Carnegie Mellon University, October 1993.

名前

polkit-is-privileged – PolicyKit 特権の検査

形式

`polkit-is-privileged [-hvV] -u` *user* `-p` *privilege* `[-r` *resource*`]`

機能説明

`polkit-is-privileged` コマンドは、システムポリシーに照会して、指定した特権およびリソースに対しユーザーが許可されているかどうかを判断します。リソース名は省略できます。Solaris オペレーティングシステムでは、特権名として RBAC 承認名を使用するようにしてください。

現在、PolicyKit の唯一のコンシューマは `hald(1M)` です。

オプション

次のオプションがサポートされています。

`-h`、`--help`	オプションの一覧を表示して終了します。
`-p` *privilege*、`--privilege` *privilege*	ユーザーに関連付けられた特権の名前。コマンドはこの特権について検査します。
`-r` *resource*、`--resource` *resource*	ユーザーおよび特権に関連付けられたリソースの名前。コマンドはこのリソースについて検査します。
`-u` *user*、`--user` *user*	検査する対象のユーザー名またはユーザー ID。
`-v`、`--verbose`	冗長モード。
`-V`、`--version`	バージョン番号を表示します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library/policykit
インタフェースの安定性	流動的

関連項目

`auths(1)`, `profiles(1)`, `hald(1M)`, `getauthattr(3C)`, `auth_attr(4)`, `policy.conf(4)`, `prof_attr(4)`, `user_attr(4)`, `attributes(5)`

名前

pooladm – リソースプール機能の有効化および無効化

形式

```
/usr/sbin/pooladm [-n] [-s] [-c] [filename] | -x
/usr/sbin/pooladm [-d | -e]
```

機能説明

pooladm コマンドは、プールやセットに対する管理操作を行います。pooladm は、指定された名前のファイルを読み取り、そこに含まれているプール構成を有効にします。

pooladm は、現在のプール実行時構成を更新する前に、その構成が正しいか検査します。

オプションを指定しないで実行すると、pooladm は、現在実行中のプール構成を表示します。

オプション

次のオプションがサポートされています。

- -c　指定された場所にある構成をインスタンス化します。ファイル名が指定されなかった場合のデフォルトは、/etc/pooladm.conf になります。
- -d　プール機能を無効にして、プールを操作できないようにします。
- -e　プール機能を有効にして、プールを操作できるようにします。
- -n　現在のアクティブ構成を実際に更新することなく構成を検査します。構文エラーがないことと、現在のシステム上で構成をインスタンス化できることを検査します。アプリケーション固有のプロパティーの検査は実行されません。
- -s　指定された場所を現在の動的構成の内容で更新します。

  このオプションを指定した場合、更新する構成に対する更新権限が必要となります。このオプションを -c オプションと併用した場合、動的な構成が更新されてから、静的な場所が更新されます。
- -x　現在アクティブになっているプール構成を削除します。定義されたすべてのリソースを破棄し、それまで区分化されていたすべてのコンポーネントを、それぞれのデフォルトリソースに戻します。

オペランド

次のオペランドがサポートされています。

filename　このファイル内に格納されている構成を使用します。

使用例

例 1　構成をインスタンス化する

次のコマンドは、/home/admin/newconfig に保存されている構成をインスタンス化します。

```
example# /usr/sbin/pooladm -c /home/admin/newconfig
```

例 2 インスタンス化することなく構成を検査する

次のコマンドは、/home/admin/newconfig に保存されている構成をインスタンス化しようとします。発生したエラー条件はすべて表示しますが、アクティブ構成を実際に変更することはありません。

```
example# /usr/sbin/pooladm -n -c /home/admin/newconfig
```

例 3 現在の構成を削除する

次のコマンドは、現在のプール構成を削除します。

```
example# /usr/sbin/pooladm -x
```

例 4 プール機能を有効にする

次のコマンドは、プール機能を有効にします。

```
example# /usr/sbin/pooladm -e
```

例 5 SMF を使ってプール機能を有効にする

次のコマンドは、サービス管理機能を使ってプール機能を有効にします。smf(5) を参照してください。

```
example# /usr/sbin/svcadm enable svc:/system/pools:default
```

例 6 アクティブ構成を指定された場所に保存する

次のコマンドは、アクティブ構成を /tmp/state.backup に保存します。

```
example# /usr/sbin/pooladm -s /tmp/state.backup
```

ファイル

/etc/pooladm.conf　pooladm の構成ファイル。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/resource-mgmt/resource-pools
インタフェースの安定性	下記を参照。

呼び出しは「確実」です。出力は「不確実」です。

関連項目

poolcfg(1M), poolbind(1M), psrset(1M), svcadm(1M), pset_destroy(2), libpool(3LIB), attributes(5), smf(5)

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

注意事項

スケジューリングクラスなど、区分化可能なリソースへのバインディングのかたちで存在していないリソースバインドは、pooladm -x 操作で必ずしも変更されるとは限りません。

デフォルトでは、プール機能は Solaris 起動時に有効になりません。pooladm -e は、プール機能を明示的に有効にします。プールが有効になると、プロセッサ区分化とプロセスバインドに関する特定の API の動作が変更されます。libpool(3LIB) を参照してください。

プロセッサセットが作成されたシステム上でプール機能を有効にすることはできません。プール機能を有効にする前に、psrset(1M) コマンドまたは pset_destroy(2) を使ってプロセッサセットを手動で破棄してください。

リソースプール機能は smf(5) サービスであるため、その有効化および無効化を標準の SMF インタフェースを使って行うこともできます。

名前

poolbind – bind processes, tasks, or projects or query binding of processes to resource pools

形式

```
/usr/sbin/poolbind -p poolname -e command [arguments]...
/usr/sbin/poolbind -p poolname [-i idtype] id...
/usr/sbin/poolbind -q pid...
/usr/sbin/poolbind -Q pid...
```

機能説明

The `poolbind` command allows an authorized user to bind zones, projects, tasks, and processes to pools. With the `-e` option (see below), it can execute a command you specify, placing the executed command in a specified pool. It can also enable you to query a process to determine which pool a process is bound to.

オプション

The following options are supported:

`-e` *command* [*arguments*...]
: Executes *command*, bound to the pool you specify with `-p`.

`-i` *idtype*
: This option, together with the *idlist* arguments, specifies one or more processes to which the poolbind command is to apply. The interpretation of *idlist* depends on the value of *idtype*. The valid *idtype* arguments and corresponding interpretations of *idlist* are as follows:

  pid
  : *idlist* is a list of process IDs. Binds the specified processes to the specified pool. This is the default behavior if no idtype is specified.

  taskid
  : *idlist* is a list of task IDs. Bind all processes within the list of task IDs to the specified pool.

  projid
  : *idlist* is a list of project IDs. Bind all processes within the list of projects to the specified pool. Each project ID can be specified as either a project name or a numerical project ID. See `project(4)`.

  zoneid
  : *idlist* is a list of zone IDs. Bind all processes within the list of zones to the specified pool. Each zone ID can be specified as either a zone name or a numerical zone ID. See `zones(5)`.

`-p` *poolname*
: Specifies the name of a pool to which the specified zone, project, tasks, or processes are to be bound.

`-q` *pid* ...
: Queries the pool bindings for a given list of process IDs. If the collection of resources associated with the process does not correspond to any currently existing pool, or if there are multiple pools with the set of resources that the process is bound to, the query fails for that particular process ID.

`-Q` *pid* ...	Queries the resource bindings for a given list of process IDs. The resource bindings are each reported on a separate line.

使用例

例 1 Binding All Processes

The following command binds all processes in projects 5 and 7 to the pool `web_app`:

```
example# /usr/sbin/poolbind -p web_app -i projid 5 7
```

例 2 Binding the Running Shell

The following command binds the running shell to the pool `web_app`:

```
example# /usr/sbin/poolbind -p web_app $$
```

例 3 Querying the Pool Bindings

The following command queries the bindings to verify that the shell is bound to the given pool:

```
example# /usr/sbin/poolbind -q $$
```

例 4 Querying the Resource Bindings

The following command queries the bindings to verify that the shell is bound to the given resources:

```
example# /usr/sbin/poolbind -Q $$
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Requested operation could not be completed.

`2` Invalid command line options were specified.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/resource-mgmt/resource-pools
Interface Stability	See below.

The invocation is Committed. The output is Uncommitted.

関連項目

`pooladm(1M)`, `poolcfg(1M)`, `libpool(3LIB)`, `project(4)`, `attributes(5)`, `zones(5)`

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

名前

poolcfg – リソースプール構成ファイルの作成および変更

形式

```
/usr/sbin/poolcfg -c command [-d | [filename]]
/usr/sbin/poolcfg -f command_file [-d | [filename]]
/usr/sbin/poolcfg -h
```

機能説明

poolcfg ユーティリティーは、プールやセットに対する構成操作を行います。これらの操作は既存の構成に対して実行され、「指定された構成ファイルを変更する」という方法を取ります。-d オプションを使用した場合、カーネル状態に対して変更が加えられます。結果として得られた構成を実際に有効化するには、pooladm(1M) ユーティリティーを使用します。

プール構成ファイルは、poolcfg 自体を使ったか、あるいは libpool(3LIB) を使って直接構築された構造化ファイルです。

プールの動的な場所と書き込み権を指定した poolcfg の呼び出しは、その動的な場所が書き込みのためにすでに開かれているとハングアップします。

このユーティリティーによって作成された構成は、pooladm がターゲットホスト上でこの構成をインスタンス化するために使用できます。

オプション

次のオプションがサポートされています。

-c *command*	*command* に特定の編集コマンドを指定します。「使用法」を参照してください。
-d	カーネル状態に対して直接操作を行います。*filename* は使用できません。
-f *command_file*	*command_file* からコマンドを取得します。*command_file* はいくつかの編集コマンドから構成されます。コマンドは 1 行に 1 つずつ記述されます。
-h	編集コマンドの構文に関する詳細情報を表示します。

使用法

スクリプト

スクリプトは、複数の編集コマンドで構成されます。スクリプト内の各行に、編集コマンドを 1 つずつ記述します。次のような書式を使用します。

info [*entity-name*]
: 構成 (または指定された部分) を、人間が読める形式で標準出力に表示します。エンティティーが指定されなかった場合、システムの情報が表示されます。したがって、poolcfg -c 'info' afile は、poolcfg -c 'info system name' afile と同等の呼び出しになります。

create *entity-name* [*property-list*]
: 指定されたタイプと名前のエンティティーを作成します。

destroy *entity-name*
指定されたエンティティーを削除します。

modify *entity-name* [*property-list*]
指定されたエンティティー上の指定された一連のプロパティーを変更します。

associate *pool-name* [*resource-list*]
特定のプールに 1 つ以上のリソースを接続するか、1 つ以上の既存の接続を置き換えます。

transfer to [*resourcetype*] *name*[*component-list*]
1 つ以上の個別コンポーネントを特定のリソースに転送します。

transfer [*quantity*] from [*resourcetype*] [*src*] to [*tgt*]
特定数のリソースを *src* から *tgt* に転送します。

transfer [*quantity*] to [*resourcetype*] [*tgt*] from [*src*]
特定数のリソースを *src* から *tgt* に転送します。

discover
システムエンティティーを 1 つ作成します。このエンティティーには、1 つのプールエンティティーと現在のシステム構成に一致するリソースが格納されます。発見されたすべてのリソースタイプのすべてのリソースがファイル内に記録されます。その際、単一のプールが各リソースタイプのデフォルトリソースを参照するように記録されます。

`poolcfg` がカーネルに対して直接操作を行う場合、このコマンドは何も行いません。`-d` オプションを参照してください。

このコマンドはできるだけ使用しないでください。構成を作成するために推奨されている方法は、`pooladm(1M)` の `-s` オプションを使って動的構成をエクスポートすることです。

rename *entity-name* to *new-name*
システム上の特定のエンティティーの名前を別の名前に変更します。

プロパティーリスト

プロパティーリストは次のように指定します。

```
( proptype name = value [ ; proptype name = value ]* )
```

ここで、特定のプロパティータイプと名前のペアについて、シーケンス内で最後に定義されたものが有効となります。プロパティーを削除するには、`~ proptype name` を使用します。

リソースリスト

リソースリストは次のように指定します。

```
( resourcetype name [ ; resourcetype name ]* )
```

ここで、特定のリソースについて、シーケンス内で最後に指定されたものが有効となります。リソースリストでは、削除用の構文は存在しません。

コンポーネントリスト

コンポーネントリストは次のように指定します。

```
( componenttype name [ ; componenttype name ]* )
```

ここで、特定のコンポーネントについて、シーケンス内で最後に指定されたものが有効となります。コンポーネントリストでは、削除用の構文は存在しません。

認識されるエンティティー

system　マシンレベルのエンティティー

pool　リソース関連付けの集合に名前を付けたもの

リソースタイプ

pset　プロセッサセットリソース

プロパティータイプ

boolean　2 つの値 true、false のいずれかを取ります。

int　64 ビット符号付き整数値。

uint　64 ビット符号なし整数値。

string　文字列は引用符 (") で区切ります。formats(5) で定義されている文字エスケープシーケンスがサポートされます。

float　科学用の表記はサポートされません。

使用例

例 1　poolcfg スクリプトの記述

次の poolcfg スクリプトは、Accounting という名前のプールと、プロセッサセット small-1 を作成します。まず、プロセッサセットが作成されます。次に、プールが作成され、それにプロセッサセットが関連付けられます。

```
create pset small-1 ( uint pset.min = 1 ; uint pset.max = 4)
create pool Accounting
associate pool Accounting ( pset small-1 )
```

例 2　pool_0 のレポート

次のコマンドは、pool_0 に関するレポートを、人間が読める形式で標準出力に出力します。

```
 # poolcfg -c 'info pool pool_0' /etc/pooladm.conf
```

例 3　pool_0 とその関連付けの削除

次のコマンドは、pool_0 とその関連付けを削除します。ただし、すでに関連付けられていたリソースは削除しません。

```
# poolcfg -c 'destroy pool pool_0' /etc/pooladm.conf
```

例 4　現在の構成の表示

次のコマンドは現在の構成を表示します。

```
$ poolcfg -c 'info' /etc/pooladm.conf
system example_system
               int system.version 1
```

例 4　現在の構成の表示　　*(続き)*

```
boolean system.bind-default true
string system.comment Discovered by libpool

pool pool_default
     boolean pool.default true
     boolean pool.active true
     int pool.importance 5
     string pool.comment
     string.pool.scheduler FSS
     pset pset_default

pset pset_default
     int pset.sys_id -1
     string pset.units population
     boolean pset.default true
     uint pset.max 4294967295
     uint pset.min 1
     string pset.comment
     boolean pset.escapable false
     uint pset.load 0
     uint pset.size 2

     cpu
          int cpu.sys_id 0
          string cpu.comment

     cpu
          int cpu.sys_id 2
          string cpu.comment
```

例 5　カーネル内で ID 2 の cpu をプロセッサセット pset1 に移動する

次のコマンドは、カーネル内で、ID 2 の cpu をプロセッサセット pset1 に移動します。

```
# poolcfg -dc 'transfer to pset pset1 ( cpu 2 )'
```

例 6　カーネル内で 2 つの cpu をプロセッサセット pset1 からプロセッサセット pset2 に移動する

次のコマンドでは、カーネル内で、2 つの cpu をプロセッサセット pset1 からプロセッサセット pset2 に移動します。

```
# poolcfg -dc 'transfer 2 from pset pset1 to pset2'
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/resource-mgmt/resource-pools
インタフェースの安定性	下記を参照。

呼び出しは「確実」です。出力は「不確実」です。

関連項目

pooladm(1M), poolbind(1M), libpool(3LIB), attributes(5), formats(5)

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

名前

poold – automated resource pools partitioning daemon

形式

`poold [-l level]`

機能説明

`poold` provides automated resource partitioning facilities. `poold` can be enabled or disabled using the Solaris Service Management Facility, `smf(5)`. `poold` requires the Resource Pools facility to be active in order to operate.

The dynamic resource pools service's fault management resource identifier (FMRI) is:

```
svc:/system/pools/dynamic
```

The resource pools service's FMRI is:

```
svc:/system/pools
```

`poold`'s configuration details are held in a `libpool(3LIB)` configuration and you can access all customizable behavior from this configuration.

`poold` periodically examines the load on the system and decides whether intervention is required to maintain optimal system performance with respect to resource consumption. `poold` also responds to externally initiated (with respect to `poold`) changes of either resource configuration or objectives.

If intervention is required, `poold` attempts to reallocate the available resources to ensure that performance objectives are satisfied. If it is not possible for `poold` to meet performance objectives with the available resources, then a message is written to the log. `poold` allocates scarce resources according to the objectives configured by the administrator. The system administrator must determine which resource pools are most deserving of scarce resource and indicate this through the importance of resource pools and objectives.

オプション

The following options are supported:

`-l` *level* Specify the vebosity level for logging information.

Specify *level* as `ALERT`, `CRIT`, `ERR`, `WARNING`, `NOTICE`, `INFO`, and `DEBUG`. If *level* is not supplied, then the default logging level is `INFO`.

`ALERT`	A condition that should be corrected immediately, such as a corrupted system database.
`CRIT`	Critical conditions, such as hard device errors.
`ERR`	Errors.
`WARNING`	Warning messages.
`NOTICE`	Conditions that are not error conditions, but that may require special handling.
`INFO`	Informational messages.

DEBUG　Messages that contain information normally of use only when debugging a program.

When invoked manually, with the `-l` option, all log output is directed to standard error.

使用例

例 1　Modifying the Default Logging Level

The following command modifies the default logging level to ERR:

```
# /usr/lib/pool/poold -l ERR
```

例 2　Enabling Dynamic Resource Pools

The following command enables dynamic resource pools:

```
# /usr/sbin/svcadm enable svc:/system/pools/dynamic
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/resource-mgmt/resource-pools
Interface Stability	See below.

The invocation is Committed. The output is Uncommitted.

関連項目

`pooladm(1M)`, `poolbind(1M)`, `poolcfg(1M)`, `poolstat(1M)`, `svcadm(1M)`, `pool_set_status(3POOL)`, `libpool(3LIB)`, `attributes(5)`, `smf(5)`

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

名前

poolstat – report active pool statistics

形式

```
poolstat [-p pool-list] [-r rset-list] [-T u | d ] [interval [count]]

poolstat [-p pool-list] [-o format -r rset-list]
     [-T u | d ] [interval [count]]
```

機能説明

The `poolstat` utility iteratively examines all active pools on the system. It reports statistics based on the selected output mode. `poolstat` provides options to examine only specified pools and report resource set-specific statistics.

Without options, `poolstat` examines all pools, reports basic statistics for their resource sets, and exits.

DISPLAY FORMATS

In default output format, `poolstat` outputs a header line and a line for each pool. The line begins with the pool ID and its name, followed by a column of statistical data for the processor set attached to the pool.

The columns are defined as follows:

`id` Pool ID.

`pool` Pool name.

`rid` Resource set id.

`rset` Resource set name.

`type` Resource set type.

`min` Minimum resource set size.

`max` Maximum resource set size.

`size` Current resource set size.

`used` The measure of how much of the resource set is currently is use. This is calculated as the percentage utilization of the resource set multiplied by its size. If resource set has been reconfigured during last sampling interval, this value might be not reported (`-`).

`load` The absolute representation of the load that is put on the resource set. For the definition of this property see `libpool(3LIB)`.

オプション

The following options are supported:

`-o` *format* Report statistics according to the format specification given in format. See `DISPLAY FORMATS`.

The `-o` option accepts lists as arguments. Items in a list can be either separated by commas or enclosed in quotes and separated by commas or spaces.

You can specify multiple `-o` options. The format specification is interpreted as the whitespace separated concatenation of all the format option arguments.

The -o option must be used in conjunction with the -r option.

-p *pool-list* Report only pools whose names are in the given list. If the -r option is also used, this option selects only resource sets which belong to pools in the given list. Statistics for pools or resource sets are reported in the same order in which pool names are listed on the pool-list. Pool can be specified by name or by ID.

The -p option accepts lists as arguments. Items in a *pool-list* can only be separated by spaces.

-r *rset-list* Report resource set statistics. If the *rset-list* argument is "all", then all possible resource set types are selected.

The -r option accepts lists as arguments. Items in a list can be either separated by commas or enclosed in quotes and separated by commas or spaces.

The following resource set types are supported:

all All resource set types

pset Processor set

-T u | d Display a time stamp.

Specify u for a printed representation of the internal representation of time. See time(2). Specify d for standard date format. See date(1).

オペランド

The following operands are supported:

count The number of times that the statistics are repeated. By default, poolstat reports statistics only once.

If neither interval nor count are specified, statistics are reported once. If interval is specified and count is not, statistics are reported indefinitely.

interval The sampling interval in seconds.

If neither interval nor count are specified, statistics are reported once. If interval is specified and count is not, statistics are reported indefinitely.

使用例

例 1 Using poolstat

The following example shows the default output from the poolstat utility:

```
% poolstat
                          pset
  id pool           size used load
   0 pool_default      4  3.6  6.2
   1 pool_admin        4  3.3  8.4
```

例 2 Reporting Resource Set Statistics

The following example reports resource set statistics.

```
% poolstat -r pset
  id pool           type rid rset          min  max size used load
  0 pool_default    pset  -1 pset_default    1  65K    2  1.2  8.3
  1 pool_admin      pset   1 pset_admin      1    1    1  0.4  5.2
  2 pool_other      pset  -1 pset_default    1  65K    2  1.2  8.3
```

Resource sets attached to multiple pools, as `pset_default` in the example above, are listed multiple times, once for each pool.

例 3 Restricting the Output to the List of Pools

The following example restricts the output to the list of pools

```
% poolstat -p pool_default
                     pset
 id pool         size used load
 0 pool_default    8  5.3 10.3

% poolstat -p 'pool_admin pool_default'
                       pset
  id pool         size used load
  1 pool_admin      6  4.3  5.3
  0 pool_default    2  1.9  2.0

% poolstat -r all -p 'pool_admin pool_default'
  id pool            type rid rset          min  max size used load
  1 pool_admin      pset   1 pset_admin      1    1    1  0.9  2.3
  2 pool_default    pset  -1 pset_default    1  65K    2  2.0  2.0
```

例 4 Customizing Output

The following example customizes output:

```
% poolstat -r -o pool,rset,size,load
  pool            rset          size load
  pool_default    pset_default     4  4.5
  pool_admin      pset_admin       4  2.1
```

終了ステータス

The following exit values are returned:

0 Successful completion.

1 An error occurred.

2 Invalid command line options were specified.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/resource-mgmt/resource-pools
Interface Stability	Committed

関連項目

libpool(3LIB), attributes(5)

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

注意事項

The system ids associated with resources can change after the system reboots or the resource configuration is altered.

名前

ports – creates /dev entries and inittab entries for serial lines

形式

`/usr/sbin/ports [-r rootdir]`

機能説明

devfsadm(1M) is now the preferred command for /dev and /devices and should be used instead of ports.

The ports command creates symbolic links in the /dev/term and /dev/cua directories to the serial-port character device files in /devices and adds new entries in /etc/inittab for non-system ports found. System-board ports are given single lower-case letters for names (such as a and b) while other ports are named numerically.

ports searches the kernel device tree to find the serial devices attached to the system. It also checks /dev/term and /dev/cua to see what symbolic links to serial devices already exist. ports then performs the following:

1. Assigns new numbers (or letters for system-board ports) to ports that are attached to the system but do not have /dev/term and /dev/cua entries. The numbers or letters assigned are the lowest-unused numbers or letters.
2. Removes dangling links: links from /dev/term and /dev/cua pointing to no-longer-existing ports.
3. Creates new /dev/term and /dev/cua links for new serial devices.

If the configuration has not changed, ports exits without doing anything.

Notice to Driver Writers

ports considers devices with a node type of DDI_NT_SERIAL, DDI_NT_SERIAL_MB, DDI_NT_SERIAL_DO, or DDI_NT_SERIAL_MB_DO to be serial port devices. Devices with one of these node types must create minor device names that obey the following conventions when calling ddi_create_minor_node(9F).

- The minor name for non-system port devices (DDI_NT_SERIAL) consists of an ASCII numeric string, where the first port on the device is named 0, the second named 1, the third named 2, up to the number of ports provided by the device.
- The minor name for non-system dialout devices (DDI_NT_SERIAL_DO) is the ASCII numeric port name, concatenated with ,cu. For example, the minor name for the first dialout port on the serial board is 0,cu.
- The minor name for system-board port devices (DDI_NT_SERIAL_MB) consists of a string containing a single ASCII lowercase character, where the first port on the device is named a, the second is named b, the third is named c, for all ports on the device (or up through port z).
- The minor name for system-board dialout devices (DDI_NT_SERIAL_MB_DO) consists of the lowercase character port name, concatenated with ,cu. For example, the minor name for the first dialout port on the on-board serial device is a,cu.

To prevent disks from attempting to automatically generate links for a device, drivers must specify a private node type and refrain from using one of the above node types when calling ddi_create_minor_node(9F).

オプション

The following options are supported:

`-r` *rootdir* Causes `ports` to presume that the `/dev/term`, `/dev/cua`, and `/devices` directories are found under *rootdir*, not directly under `/`.

使用例

例 1 Creating the Serial and Dialout Minor Device Nodes

The following example creates the serial and dialout minor device nodes from the `xkserial` driver's `attach(9E)` function:

```
/*
* Create the minor number by combining the instance number
* with the port number.
*/ #define XKNUMPORTS           8
#define XKMINORNUM(i, p)     ((i) << 4 | (p))
#define XKMINORNUM_DO(i, p)  ((i) << 4 | (p) | 0x80)
int
xkserialattach(dev_info_t *dip, ddi_attach_cmd_t cmd)
{
   int instance, portnum;
   char name[8];
       /* other stuff in attach... */
   instance = ddi_get_instance(dip);
   for (portnum = 0; portnum < XKNUMPORTS; portnum++) {
       /*
        * create the serial port device
        */
        sprintf(name, "%d", portnum);
        ddi_create_minor_node(dip, name, S_IFCHR,
           XKMINORNUM(instance, portnum), DDI_NT_SERIAL, 0);

        /*
         * create the dialout device
         */
         sprintf(name,"%d,cu", portnum);
         ddi_create_minor_node(dip, name, S_IFCHR,
             XKMINORNUM_DO(instance, portnum), DDI_NT_SERIAL_DO, 0);
     }
}
```

例 2 Installing the xkserial Port Driver on a Sun Fire 4800

The following example installs the `xkserial` port driver on a Sun Fire 4800 (with the driver controlling the fictional XKSerial 8 port serial board), with these special files in `/devices`:

```
# ls -l /devices/ssm@0,0/pci@18,700000/pci@1/xkserial@f,800000/
crw-r-----   1 root sys   32,  16 Aug 29 00:02 xkserial@2000:0
crw-r-----   1 root sys   32, 144 Aug 29 00:02 xkserial@2000:0,cu
crw-r-----   1 root sys   32,  17 Aug 29 00:02 xkserial@2000:1
```

例 2 Installing the xkserial Port Driver on a Sun Fire 4800 (続き)

```
crw-r-----   1 root sys   32, 145 Aug 29 00:02 xkserial@2000:1,cu
crw-r-----   1 root sys   32,  18 Aug 29 00:02 xkserial@2000:2
crw-r-----   1 root sys   32, 146 Aug 29 00:02 xkserial@2000:2,cu
crw-r-----   1 root sys   32,  19 Aug 29 00:02 xkserial@2000:3
crw-r-----   1 root sys   32, 147 Aug 29 00:02 xkserial@2000:3,cu
crw-r-----   1 root sys   32,  20 Aug 29 00:02 xkserial@2000:4
crw-r-----   1 root sys   32, 148 Aug 29 00:02 xkserial@2000:4,cu
crw-r-----   1 root sys   32,  21 Aug 29 00:02 xkserial@2000:5
crw-r-----   1 root sys   32, 149 Aug 29 00:02 xkserial@2000:5,cu
crw-r-----   1 root sys   32,  22 Aug 29 00:02 xkserial@2000:6
crw-r-----   1 root sys   32, 150 Aug 29 00:02 xkserial@2000:6,cu
crw-r-----   1 root sys   32,  23 Aug 29 00:02 xkserial@2000:7
crw-r-----   1 root sys   32, 151 Aug 29 00:02 xkserial@2000:7,cu
```

`/dev/term` contain symbolic links to the serial port device nodes in `/devices`

```
# ls -l /dev/term
/dev/term/0 -> ../../devices/[....]/xkserial@2000:0
/dev/term/1 -> ../../devices/[....]/xkserial@2000:1
/dev/term/2 -> ../../devices/[....]/xkserial@2000:2
/dev/term/3 -> ../../devices/[....]/xkserial@2000:3
/dev/term/4 -> ../../devices/[....]/xkserial@2000:4
/dev/term/5 -> ../../devices/[....]/xkserial@2000:5
/dev/term/6 -> ../../devices/[....]/xkserial@2000:6
/dev/term/7 -> ../../devices/[....]/xkserial@2000:7
```

and `/dev/cua` contain symbolic links to the dialout port device nodes in `/devices`

```
# ls -l /dev/cua

/dev/cua/0 -> ../../devices/[....]/xkserial@2000:0,cu
/dev/cua/1 -> ../../devices/[....]/xkserial@2000:1,cu
/dev/cua/2 -> ../../devices/[....]/xkserial@2000:2,cu
/dev/cua/3 -> ../../devices/[....]/xkserial@2000:3,cu
/dev/cua/4 -> ../../devices/[....]/xkserial@2000:4,cu
/dev/cua/5 -> ../../devices/[....]/xkserial@2000:5,cu
/dev/cua/6 -> ../../devices/[....]/xkserial@2000:6,cu
/dev/cua/7 -> ../../devices/[....]/xkserial@2000:7,cu
```

ファイル

`/dev/term/`*n*	Logical serial port devices
`/dev/cua/`*n*	Logical dialout port devices
`/etc/inittab`	Controls dispatching by `init`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

add_drv(1M), devfsadm(1M), drvconfig(1M), attributes(5), devfs(7FS), attach(9E), ddi_create_minor_node(9F)

『Writing Device Drivers』

名前

poweradm – manage power management properties

形式

```
poweradm [-v] get [-a all|smf|platform|current] property ...
poweradm [-v] set property=value ...
poweradm [-v] list
poweradm -?
```

機能説明

The `poweradm` program is used to display and manage the Power Management settings within a Solaris instance.

The Power Management properties and their values are stored in the service management facility (see `smf(5)`).

All users can run the `list` and `get` forms given in the `SYNOPSIS`. These commands allow all users to view the current Power Management settings within a Solaris instance.

Only users and roles that belong to the Maintenance and Repair RBAC profile can execute the `set` form of `poweradm` given in the `SYNOPSIS`. See also the `NOTES` section below.

`poweradm` supports the following service property:

```
administrative-authority
```

The value for this property is specified as a string and can have the values:

- `smf`
- `platform`
- `none`

...with `platform` as the default value. The significance of this property is the source of administrative control for power management within the Solaris kernel. That is, this property indicates the origin of `time-to-full-capacity` and `time-to-minimum-responsiveness` settings within the Solaris kernel. Only a Solaris user with appropriate privileges can set this property.

When the `administrative-authority` is set to `platform` the values of `time-to-full-capacity` and `time-to-minimum-responsiveness` will be taken from the platform code. Setting these values in SMF using the `poweradm` command will have no effect upon the values in the kernel, while `administrative-authority` is set to `platform`. Under this condition, the `poweradm` command will indicate that the kernel is currently using the platform values. As these values are changed by the platform administrator, so they will be modified in kernel. Also, commands to enable or disable the power management within the Solaris kernel will come from the platform code. When the `administrative-authority` is set to `smf`, the values of `time-to-full-capacity` and `time-to-minimum-responsiveness` will be taken from SMF. As these values in SMF are changed by the Solaris administrator, so they will be applied to kernel. Setting these values in the platform will have no effect upon the values in the kernel while administrative-authority is set to `smf`. Under this condition, he `poweradm`

command will indicate that the kernel is currently using the values from SMF. As these values are changed by the Solaris administrator, so they will be modified in kernel.

When the `administrative-authority` is set to `none`, power management within the Solaris kernel will be turned off, although the power service will continue to run. Any power management instructions from the platform will be ignored, as will the settings in SMF. Only when `administrative-authority` is set to one of the other values will power management within the Solaris kernel restart, using the settings from the specified source.

`time-to-full-capacity`
: Specified in microseconds.

  This parameter constrains the dynamic capacity adjustment allowed while the system is in an active state.

  This parameter defines the maximum time the system is allowed to reach (re-provision and make available) its full capacity, returning from any lower-capacity/less-responsive state, while it has been using any or all of the PM features falling within this bound.

  By default, this value is taken from the platform (for example, `i86pc`), because the default setting for `administrative-authority` is set to `platform`.

  Alternatively, if `administrative-authority` is set to `smf`, this value is taken from the definition provided by the power service (that is, SMF). At install time this value is set to be `undefined`. If the Solaris administrator chooses to modify this property, a value appropriate to the needs of the workload or applications must be picked.

`time-to-minimum-responsiveness`
: Specified in milliseconds.

  This parameter constrains the dynamic capacity adjustment allowed while the system is in an inactive state.

  This parameter defines how long the system is allowed to return to its active state—that is, to provide the minimum capacity required to meet the above time-to-full-capacity constraint.

  Moderate values (seconds) allow hardware components or subsystems on the platform to be placed in slower-response inactive states; larger values still (for example, 30 seconds to minutes) allow for such as whole system suspension, using techniques such as suspend-to-RAM.

  By default, this value is taken from the platform (for example, `i86pc`), because the default setting for `administrative-authority` is set to `platform`.

  Alternatively, if `administrative-authority` is set to `smf`, this value is taken from the definition provided by the power service (that is, SMF). At install time this value is set to be `undefined`. If the Solaris administrator chooses to modify this property, a value appropriate to the needs of the workload or applications must be picked.

`suspend-enable`
By default, only a small number of machines are allowed to suspend. Setting this property to true will permit other machines to attempt to suspend. By default this property is set to `false`, unless a machine has S3 according the the ACPI kstat module and is on the white list. If these two conditions are met, the default is `true`. The value of `administrative-authority` has no effect upon this property.

オプション

The following options are supported.

`-?`
Display a synopsis of available subcommands and options.

`-v`
Provide verbose output. Can be used with any of the subcommands listed below.

サブコマンド

The following subcommands are supported:

`get [-a all|smf|platform|current]`
Retrieves the current value of the named property. The `-a` option can be used to indicate the origin of the value either: SMF (`smf`), the platform (`platform`), the current value used by the kernel (`current`) or all of the preceeding (`all`). By default, if no origin is specified then `current` is assumed.

The `administrative-authority` and `suspend-enable` properties do not have a platform value

`set` *property=value...*
Changes the named *property* to the given *value*. The properties `administrative-authority` and `suspend-enable` are automatically synchronized to the new value in the kernel. The properties `time-to-full-capacity` and `time-to-minimum-responsiveness` are synchronized to the kernel if and only if `administrative-authority` is set to `smf`. If the `-v` option is used and the kernel cannot be updated immediately because `administrative-authority` is not set to `smf`, a warning message will be issued.

Only users and roles that belong to the Maintenance and Repair RBAC profile can execute the `set` subcommand.

`list`
Lists all the available Power Management properties values and indicate whether power management is active.

終了ステータス

`0`
Successful completion.

`1`
An error occurred.

`2`
Invalid command line options were specified.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/kernel/power
Interface Stability	Committed

関連項目

attributes(5), smf(5), smf_security(5)

注意事項

The power service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/power:default
```

The properties that can be set by poweradm are defined in that service as:

- active_control/administrative-authority, which is described under administrative-authority above
- active_config/time-to-full-capacity, which is described under time-to-full-capacity above
- active_config/time-to-minimum-responsiveness, which is described under time-to-minimum-responsiveness above
- suspend/suspend-enable, which is described under suspend-enable above

If the service is disabled, no power management settings will be uploaded to the kernel in the future. Existing settings will not be undone until the next reboot. Disabling all power management is best accomplished by setting administrative-authority to none.

If administrative-authority is set to the value smf before both time-to-full-capacity and time-to-minimum-responsiveness have been set, the service will go into maintenance mode. In such a situation, set administrative-authority to the value none then set both time-to-full-capacity and time-to-minimum-responsiveness to the values you want, clear the service and then set administrative-authority to smf.

To set properties in the active_config and suspend property groups, the solaris.smf.value.power_config authorization is required. To set properties in the active_control property group requires the solaris.smf.value.power_control authorization is required. Both of these authorizations are part of the Maintenance and Repair profile.

名前

powertop – 電源管理に影響するイベントの報告と分析

形式

`powertop [-c processor_id] [-d count] [-t interval] [-v] [-h]`

機能説明

PowerTOP は、システムがどれだけ効果的に、CPU の電源管理機能を利用しているかを表示する可観測性ツールです。本来ならアイドル状態になっているシステムでツールを実行することによって、ユーザーは、別の電力状態で CPU が稼働している時間を調べられます。未使用 (アイドル) のシステムでは、そのすべての時間にわたって最低の電力ステータスで稼働することが理想的ですが、バックグラウンドのユーザーおよびカーネルアクティビティー (ステータスをポーリングするために定期的に起動するランダムなソフトウェア) のために、アイドルシステムでも必要以上に電力を消費する場合があります。

このツールは、システムアクティビティーを定期的に分析し、サポートしているそれぞれの電力状態でプロセッサが稼働している時間に関する要約を示します。また、CPU を起動させ、より多くの電力を消費させる原因となった上位アクティビティーも表示します。ユーザーはこのレポートを通じて、システムの問題領域を特定および診断し、その電力効率を最適化できます。

PowerTOP は、CPU が低い電力状態に移行することを妨げているアクティビティーの量を平均し、「Wakeups-from-idle per second」フィールドにその値を表示します。この値は、ウェイクアップの合計回数を現在の時間間隔で割ったものです。いつでもすべてのイベントが画面に表示されるわけではありません。

実行中に R キーを押すと、分析が強制的に更新されます。時間間隔は、デフォルトまたは指定した値に復元されます。アプリケーションを終了するには、Q キーを押す必要があります。

PowerTOP を使用すると、3 つのサブウィンドウをそれぞれフリーズすることができ、各サブウィンドウの内容をより良く分析できます。3 つの異なる DTrace スクリプトを同時に実行している場合、サブウィンドウをフリーズすると、CPU の消費電力が低下します。この機能は、ツールの実行中に i、f、または e キーを押すと有効になります。これらのキーのいずれかを押すと、アイドル、頻度、またはイベントレポートがそれぞれフリーズします。もう一度押すと、フリーズは解除されます。この機能は、分析されているアプリケーションの実行中に呼び出したときにもっとも有効です。

PowerTOP は、一部の仮想ドメインで動作します。ただし、アイドル状態遷移のレポートは、物理的 CPU を複数の仮想 CPU が共有する場合があるため、正確であることも、正確でないこともあります。ウェイクアップ回数とイベントレポートはどちらも、現在仮想化されている環境に関する情報を表示します。

オプション

次のオプションがサポートされています。

`-c` [*processor_id*]
ツールが監視すべき CPU を指定します。

`-d` [*count*]
システムアクティビティーの *count* 分析の結果を画面にダンプします。

-h
コマンドの使用法を表示します。

-t [*interval*]
ツールがシステムを分析する間隔を秒単位で指定します。指定可能な値は、1 から 30 の範囲です。デフォルトは 5 秒です。

-v
詳細モードに切り換えて、カーネル循環サブシステムの発動などをイベントレポートに記します。

使用例

例 1 間隔を設定する

次のコマンドは、間隔を 2 秒に設定します。

```
% powertop -t 2
```

例 2 システムアクティビティーを分析しダンプする

次のコマンドは、システムアクティビティーを分析し標準出力に 4 回ダンプします。

```
% powertop -d 4
```

例 3 循環サブシステムアクティビティーを報告する

次のコマンドは、循環サブシステムアクティビティーを報告します。

```
% powertop -v
```

例 4 特定のプロセッサでのアクティビティーを分析する

次のコマンドは、PowerTOP を実行し、CPU 3 のデータのみを表示します。

```
% powertop -c 3
```

終了ステータス

0
正常終了。

1
エラーが発生しました。

2
使用法が正しくありません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
アーキテクチャ	x86、SPARC
使用条件	diagnostic/powertop

属性タイプ	属性値
インタフェースの安定性	流動的

関連項目

kstat(1M), psrinfo(1M), uadmin(2), libdevinfo(3LIB), attributes(5), cpr(7), pm(7D), pm-components(9P), removable-media(9P)

SunOS 以外のマニュアルページでの xscreensaver(1) および dtpower(1M)。

使用法

PowerTOP を実行するには DTrace 特権が必要であり、ツールがシステムに改善を提案するには、ルート (スーパーユーザー) 特権があるか、または Primary Administrator 役割を引き受けている必要があります。

名前

pppd – ポイントツーポイントプロトコルデーモン

形式

pppd [*tty_name*] [*speed*] [*options*]

機能説明

ポイントツーポイントプロトコル (PPP) は、シリアルのポイントツーポイントリンク上でデータグラムを伝送する方法を提供します。PPP は、シリアルリンク上でのデータグラムのカプセル化機能、拡張可能なリンク制御プロトコル (LCP)、およびさまざまなネットワークレイヤープロトコルの確立と設定を行う一群のネットワーク制御プロトコル (NCP) の 3 つの部分から構成されています。

カプセル化方式は、カーネル内のドライバコードによって提供されます。pppd は、基本的な LCP 認証機能、およびインターネットプロトコル (IP) と IPv6 の確立と設定を行ういくつかの NCP (IP 制御プロトコル (IPCP) および IPV6CP と呼ばれる) を提供します。

オプション

ここでは、pppd のオプションについて説明します。

オプションファイル

オプションは、ファイルおよびコマンド行から読み込まれます。pppd は、コマンド行のオプションを処理する前に、/etc/ppp/options、$HOME/.ppprc、および /etc/ppp/options.*ttyname* ファイルからこの順にオプションを読み込みます。(options.*ttyname* ファイルの読み込み前に、端末名を取得するためにコマンド行オプションがスキャンされる。)options.*ttyname* ファイル名を構築するために、端末名から先頭の /dev/ が取り除かれ、残ったスラッシュ文字 (/) がドットで置き換えられます。たとえば、シリアルデバイス /dev/cua/a の場合、/etc/ppp/options.cua.a が読み込まれます。

オプションファイルの内容は、スペースで区切られた一連の文字列として構文解析されます。スペースを含む文字列は二重引用符 (") で囲むことで 1 つの文字列として解釈されるようになります。バックスラッシュ (\) は、直後の 1 文字をクォートします。ハッシュ文字 (#) はコメントの始まりとして解釈され、行末までがコメントとみなされます。オプションファイル中での file および call のオプションの使用制限はありません。

よく使われるオプション

<tty_name>

指定したデバイスを介して通信します。必要に応じて、前に /dev/ 文字列が追加されます。デバイス名が指定されていない場合や標準入力に接続された端末の名前が指定された場合、pppd はその端末を使用し、バックグラウンド実行のためのフォークを行いません。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがそれを上書きすることはできません。

<speed>

ボーレートを <speed> に設定します (10 進数)。デフォルトでは、ボーレートは変更されません。通常、このオプションはダイアルアウトにのみ必要です。

asyncmap <*map*>

async 文字マップを <*map*> に設定します。このマップは、シリアル回線上でどの制御文字が正常に受信できないかを記述するものです。pppd は、これらの文字を 2 バイトのエスケープ

シーケンスとして送信するよう、相手に依頼します。引数は 32 ビットの 16 進数であり、各ビットがエスケープする文字を表しています。ビット 0 (00000001) は文字 0x00 を表し、ビット 31 (80000000) は文字 0x1f つまり '^_' を表します。複数の asyncmap オプションが指定された場合、それらの値の論理和が採用されます。asyncmap オプションが指定されていない場合、pppd は値 0 のネゴシエーションを試みます。相手が同意した場合、標準の制御文字はエスケープされなくなります。ネゴシエーションを無効にし、すべての制御文字をエスケープするには、default-asyncmap オプションを使用します。

auth

ネットワークパケットの送受信を許可する前に、相手に自己認証を要求します。システムにデフォルト経路が設定されている場合、このオプションはデフォルトです。auth または noauth オプションが指定されていない場合、pppd は、システムでまだ経路設定されてない IP アドレスだけを使用することを相手に許可します。

call *name*

ファイル /etc/ppp/peers/*name* からオプションを読み込みます。pppd が root 以外によって実行される場合でも、このファイルには noauth などの特権オプションを含むことができます。文字列 *name* はスラッシュ (/) で始まってはいけません。また、パス名の一部に連続するピリオド ("..") を含めることはできません。

callback *number*

Microsoft CBCP を使用して指定の電話番号にコールバックするように要求します。

connect *script*

script で指定された実行可能ファイルまたはシェルコマンドを使用して、シリアル回線をセットアップします。通常、このスクリプトでは chat(1M) プログラムを使用してモデムにダイアルコマンドを送り、リモート PPP セッションを開始します。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがそれを上書きすることはできません。

crtscts

シリアルポートのデータフロー制御にハードウェアフロー制御 (RTS/CTS) を使用します。crtscts、nocrtscts、cdtrcts、または nocdtrcts オプションが指定されていない場合、シリアルポートのハードウェアフロー制御の設定は変更されずにそのままになります。一部のシリアルポートでは、実際の RTS 出力がなく、このモードを使用して単方向のフロー制御を実装します。シリアルポートは、モデムから CTS によって要求されると伝送を中断しますが、コンピュータへの送信を停止するようにモデムに要求することはできません。このモードでは、DTR をモデム制御回線として使用できます。

`defaultroute`	IPCP ネゴシエーションが正常に完了すると、相手をゲートウェイとするデフォルト経路をシステムのルーティングテーブルに追加します。このエントリは、`PPP` 接続が切断された際に削除されます。`nodefaultroute` オプションが指定されている場合、このオプションは特権オプションとなります。
`disconnect` *script*	`pppd` が接続を切断したあとに、*script* で指定された実行可能コマンドまたはシェルコマンドを実行します。通常、このスクリプトは、ハードウェアモデム制御信号を使用できない場合にモデムを切断するために使用されます。モデムがすでに切断されている場合には、`disconnect` は実行されません。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがそれを上書きすることはできません。
`escape` *xx,yy,...*	伝送時にエスケープする文字を指定します。相手が `async` 制御文字マップでエスケープを要求しているかどうかには影響されません。エスケープする文字は、コンマで区切った 16 進数のリストで指定します。制御文字しか指定できない `asyncmap` オプションとは異なり、`escape` オプションではほとんどの文字を指定できます。ただし、16 進表記で 0x20 から 0x3f までと 0x5e の文字はエスケープできません。
`file` *name*	オプションをファイル *name* から読み込みます。このオプションをコマンド行または `$HOME/.ppprc` で使用する場合、`pppd` を起動するユーザーが読み取ることのできるファイルでなければなりません。このオプションにかかわらず常に `pppd` で読み込まれるファイルのリストについては、オプションファイルを参照してください。
`init` *script*	*script* で指定された実行可能コマンドまたはシェルコマンドを実行して、シリアル回線を初期化します。通常、このスクリプトでは `chat(1M)` プログラムを使用してモデムを設定し、自動応答を有効にします。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがそれを上書きすることはできません。
`lock`	シリアルデバイスに対する排他的アクセスを確実に行うために、UUCP 形式のロックファイルを作成するよう `pppd` に指示します。
`mru` *n*	最大受信ユニット (MRU) 値を *n* に設定します。`pppd` は、相手に *n* バイトを超えるパケットを送信しないよう要求します。最小の MRU 値は 128 です。デフォルトの MRU 値は 1500 です。低速のリンクでは 296 を推奨します (TCP/IP ヘッダー 40 バイト + データ 256 バイト)。IPv6 の場合、MRU は 1280 以上でなければなりません。

`mtu` *n*
: 最大転送単位 (MTU) 値を *n* に設定します。相手が MRU ネゴシエーションを通じてこれより小さい値を要求してこないかぎり、`pppd` は、PPP ネットワークインタフェースを通して送信するデータパケットのサイズを *n* バイト以下にするよう、カーネルのネットワークコードに要求します。IPv6 の場合、MTU は 1280 以上でなければなりません。

`passive`
: LCP で「passive」オプションを有効にします。このオプションを指定すると、接続を開始しようとしても相手から応答がない場合、`pppd` は有効な LCP パケットを受動的に待ち続けます。このオプションを指定しなければ、相手から応答がない場合に `pppd` は終了します。

オプション

`<local_IP_address>:<remote_IP_address>`
: ローカルインタフェースとリモートインタフェースの IP アドレスを設定します。どちらか一方を省略することもできますが、コロンは必要です。IP アドレスは、ホスト名または 10 進数ドット表記 (例: `:10.1.2.3`) で指定します。`noipdefault` オプションが指定された場合を除き、デフォルトのローカルアドレスは、そのシステムの最初の IP アドレスです。リモートアドレスは、オプションで指定されていない場合には相手から取得されます。したがって、単純な場合にはこのオプションは不要です。このオプションでローカルまたはリモートの IP アドレスを指定すると、IPCP ネゴシエーションで相手がこの指定と異なるアドレスを送ってきた場合、`pppd` はこれを拒否します。ただし、`ipcp-accept-local` や `ipcp-accept-remote` オプションが指定されている場合にはこの限りではありません。

`allow-fcs` *fcs-type*
: 相手への送信データに使用してもよい FCS タイプを設定します。*fcs-type* には、「crc16」、「crc32」、「null」、または整数をコンマで区切って指定します。デフォルトでは、既知のタイプすべてが許可されています。このオプションを指定すると、その指定に含まれていないタイプを相手が要求した場合、指定のタイプだけを要求する LCP Configure-Nak が送信されます。

`allow-ip` *address(es)*
: 相手が自己認証を行わずに使用してもよい IP アドレスまたはサブネットを指定します。このパラメータの構文解析方法は、秘密情報ファイルで許可される IP アドレスのリストの各要素が構文解析される方法と同じです。詳細については、認証の節を参照してください。

`bsdcomp` *nr,nt*
: 相手に、BSD-Compress 方式を使用して送信パケットを最大コードサイズ *nr* ビットで圧縮することを要求し、相手への送信パケットを最大コードサイズ *nt* ビットで圧縮することに同意します。*nt* が指定されていない場合は、デフォルトで *nr* の値が使用されます。*nr* と *nt* には、9 から 15 の範囲の値を使用できます。より大きな値を使用すると、圧縮率は向上しますが、圧縮辞書のためにより多くのカーネルメモリーが消費されます。*nr* または *nt* に値 0 を指定すると、その方向の圧縮は無効になります。BSD-Compress 圧縮を完全に無効にするに

は、nobsdcomp か bsdcomp 0 を指定してください。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがその値を超えるコードサイズを指定することはできません。

cdtrcts
シリアルポートのデータフロー制御に標準でないハードウェアフロー制御 (DTR/CTS など) を使用します。crtscts、nocrtscts、cdtrcts、または nocdtrcts オプションが指定されていない場合、シリアルポートのハードウェアフロー制御の設定は変更されずにそのままになります。一部のシリアルポートには、実際の RTS 出力がありません。そのようなシリアルポートでは、このモードを使用して双方向のフロー制御を実装します。ただし、このフロー制御モードでは、DTR をモデム制御回線として使用できません。

chap-interval *n*
このオプションが指定された場合、pppd は *n* 秒に 1 度、相手へのチャレンジを繰り返します。

chap-max-challenge *n*
CHAP チャレンジの最大転送回数を *n* 回に設定します (デフォルト値は 10)。

chap-restart *n*
CHAP 再開の間隔 (チャレンジの再送のタイムアウト) を *n* 秒に設定します。デフォルトは 3 です。

connect-delay *n*
接続スクリプトが終了したあと、相手からの有効な PPP パケットを *n* ミリ秒まで待機します。この待機時間が経過するか、相手から有効な PPP パケットが受信されると、pppd は最初の LCP パケットを送信してネゴシエーションを開始します。デフォルト値は 1000 (1 秒) です。待機時間が適用されるのは、connect または pty オプションの使用時のみです。

datarate *n*
pty、notty、record、または socket オプションの使用時の最大データ速度を *n* (バイト/秒) に設定します。

debug
接続のデバッグ機能を有効にします。このオプションを指定した場合、pppd は送受信したすべての制御パケットの内容を可読形式でログに記録します。パケットは、daemon 機能を使用し、syslog を経由して debug レベルで記録されます。この情報は、/etc/syslog.conf を適切に設定することによって、ファイルに記録できます。

default-asyncmap
asyncmap ネゴシエーションを無効にし、すべての制御文字を送受信両方向でエスケープさせます。

`default-fcs`
FCS 代替ネゴシエーションを完全に無効にします。デフォルトでは、FCS 代替オプションは相手に送信されませんが、オプションは受け入れられます。相手がこのオプションを指定した場合は、LCP Configure-Reject が送信されます。

`default-mru`
MRU (最大受信ユニット) ネゴシエーションを無効にします。このオプションを指定すると、`pppd` はデフォルトの MRU 値 1500 バイトを送受信両方向で使用します。

`deflate` *nr,nt,e*
相手に、`deflate` 方式を使用して送信パケットを最大ウィンドウサイズ *2**nr* バイトで圧縮することを要求し、相手への送信パケットを最大ウィンドウサイズ *2**nt* バイト、エフォートレベル *e* (1 から 9) で圧縮することに同意します。*nt* が指定されていない場合は、デフォルトで *nr* の値が使用されます。*e* を指定しないと、そのデフォルトは 6 になります。*nr* と *nt* には、9 から 15 の範囲の値を使用できます。より大きな値を使用すると、圧縮率は向上しますが、圧縮辞書のためにより多くのカーネルメモリーが消費されます。(zlib のバグのため、値 8 は使用不可。) *nr* または *nt* に値 0 を指定すると、その方向の圧縮は無効になります。`deflate` 圧縮を完全に無効にするには、`nodeflate` か `deflate 0` を指定してください。(注: 相手がどちらにも対応できる場合、`pppd` は BSD-Compress よりも deflate 圧縮を優先して要求する。)このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがその値を超えるコードサイズを指定することはできません。

`demand`
要求時のみ、つまりデータトラフィックが存在する場合のみ、リンクを開始します。このオプションを使用する場合、リモート IP アドレスを、ユーザーがコマンド行で指定するか、オプションファイルで指定する必要があります。`pppd` は、相手に接続することなく、まず IP トラフィック用にインタフェースを設定して使用可能にします。トラフィックが現れると、`pppd` は相手に接続し、ネゴシエーションや認証などの処理を実行します。これが完了すると、`pppd` はリンクを介してデータパケットの送受信を開始します。`demand` オプションは暗黙的に `persist` オプションを指定します。この動作が望ましくない場合、`demand` オプションのあとに `nopersist` オプションを使用してください。`idle` および `holdoff` オプションも `demand` オプションとともに使用できます。

`domain` *d*
認証のために使用するローカルホスト名にドメイン名 *d* を付加します。たとえば、`gethostname()` で `porsche` という名前が返される場合に、完全指定のドメイン名が `porsche.Quotron.COM` であれば、`domain Quotron.COM` と指定できます。このように設定すると、`pppd` は、秘密情報ファイルの中の秘密情報を調べる場合や、相手に対して自己認証を行うときのデフォルト名として、`porsche.Quotron.COM` を使用します。このオプションは特権オプションです。

`endpoint` *endpoint-value*
エンドポイント判別子を設定します (通常は RFC 1990 マルチリンク PPP 処理に使用される)。*endpoint-value* は、クラス識別子とクラスに依存する値で構成されます。クラス識別子
は、「null」、「local」、「IP」、「MAC」、「magic」、「phone」、または 10 進整数のいずれかです。クラスに依存する値が存在する場合は、コロン (:) またはピリオド (.) で識別子から区切ります。この値には、「IP」クラスの場合は標準のドット区切り 10 進表記 IP アドレス、「MAC」クラスの場合は (必要に応じてコロンまたはドットで区切った) 16 進数 6 桁の Ethernet アドレス、あるいは、必要に応じてコロンまたはドットでバイトを区切った 16 進表記の任意のバイト文字列を指定できます。このオプションは使用可能ですが、この実装はマルチリンクをサポートしていません。

`fcs` *fcs-type*
相手からの送信データに使用してもらいたい FCS タイプを設定します。*fcs-type* には、「`crc16`」、「`crc32`」、「`null`」、または整数をコンマで区切って指定します。デフォルトでは、FCS 代替オプションは指定されず、媒体に依存する FCS タイプが使用されます。このオプションを指定した場合、相手が LCP Configure-Nak を送信したときは、指定したタイプだけが使用されます。共通のタイプがない場合、FCS 代替オプションは次の LCP Configure-Request から省略され、デフォルトに戻ります。

`hide-password`
このオプションを指定すると、`pppd` は PAP パケットの内容をログに記録するときにパスワード文字列を除外します。これはデフォルトです。

`holdoff` *n*
リンクの切断から再開まで何秒待つかを指定します。`persist` または `demand` オプション使用時にのみ、このオプションは有効です。アイドルであったためにリンクが切断された場合は、この抑止期間は適用されません。

`ident` *string*
LCP 識別文字列を設定します。デフォルト値は、`--version` オプションで表示されるものに似たバージョン文字列です。

`idle` *n*
リンクのアイドル状態が *n* 秒続いた場合は切断するように `pppd` に指定します。データパケット (つまり IP パケット) の送受信が行われていないとき、リンクはアイドル状態とみなされます。`demand` オプションを使用せずに `persist` オプションを使用する場合は、このオプションを使用しないでください。

`ipcp-accept-local`
このオプションを指定すると、別のオプションでローカル IP アドレスが指定されている場合でも、`pppd` は相手からのローカル IP アドレスの指定を受け入れます。

`ipcp-accept-remote`
このオプションを指定すると、別のオプションでリモート IP アドレスが指定されている場合でも、`pppd` は相手からのリモート IP アドレスの指定を受け入れます。

`ipcp-max-configure` *n*
IPCP Configure-Request の最大転送回数を *n* 回に設定します (デフォルト値は 10)。

`ipcp-max-failure` *n*
IPCP Configure-Reject の送信を開始するまでの IPCP Configure-NAK の最大送信回数を *n* 回に設定します (デフォルト値は 10)。

`ipcp-max-terminate` *n*
IPCP Terminate-Request の最大転送回数を *n* 回に設定します (デフォルト値は 3)。

`ipcp-restart` *n*
IPCP 再開の間隔 (再送のタイムアウト) を *n* 秒に設定します (デフォルト値は 3)。

`ipparam` *string*
ip-up と ip-down スクリプト用に追加のパラメータを指定します。このオプションを指定すると、*string* がこれらのスクリプトに 6 番目のパラメータとして指定されます。スクリプトの節を参照してください。

`ipv6` <*local_interface_identifier*>,<*remote_interface_identifier*>
ローカルおよびリモートの 64 ビットのインタフェース識別子を設定します。どちらか一方を省略することもできます。識別子は、IPv6 アドレスの標準 ASCII 表記法で指定する必要があります (例: `::dead:beef`)。`ipv6cp-use-ipaddr` オプションを指定すると、ローカル識別子とリモート識別子はそれぞれ対応する IPv4 アドレスから生成されます (前記を参照)。`ipv6 <local>,<remote>` オプションの代わりに `ipv6cp-use-persistent` オプションを使用することもできます。

`ipv6cp-accept-local`
相手のインタフェース識別子をローカルリンク識別子として受け入れます。

`ipv6cp-max-configure` *n*
IPv6CP Configure-Request の最大転送回数を *n* 回に設定します (デフォルト値は 10)。

`ipv6cp-max-failure` *n*
IPv6CP Configure-Reject の送信を開始するまでの IPv6CP Configure-NAK の最大送信回数を *n* 回に設定します (デフォルト値は 10)。

`ipv6cp-max-terminate` *n*
IPv6CP Terminate-Request の最大転送回数を *n* 回に設定します (デフォルト値は 3)。

`ipv6cp-restart` *n*
IPv6CP 再開の間隔 (再送のタイムアウト) を *n* 秒に設定します (デフォルト値は 3)。

`ipv6cp-use-ipaddr`
ローカルまたはリモートの IPv6 アドレスが指定されていない場合、対応する構成済み IPv4 アドレスをデフォルトのインタフェース識別子として使用します。このオプションを指定すると、ネゴシエーションで取得されたアドレスではなく、構成済みのアドレスが使用されます。このオプションを使用する場合、ローカルの IPv6 識別子を指定しないときは `ipcp-accept-local` オプション、リモートの IPv6 識別子を指定しないときは `ipcp-accept-remote` オプションを併用しないでください。

`ipv6cp-use-persistent`
一意に使用可能な永続的値をリンクローカルアドレスに使用します。

`kdebug` *n*
カーネルレベルの PPP ドライバのデバッグコードを有効にします。引数 *n* には、次の値のうち必要なものの合計を指定します。`1` は一般的なデバッグメッセージ出力を有効にします。`2` は受信したパケットの内容の出力を要求します。`4` は送信したパケットの内容の出力を要求します。カーネルによって出力されるメッセージは、`/etc/syslog.conf` 設定ファイルで指定されているファイルに syslogd(1M) によって記録されます。障害の発生したリンクをデバッグする場合には、`kdebug` オプションを使用しないでください。代わりに、`debug` オプションを使用してください。

`lcp-echo-failure` *n*
このオプションを指定すると、LCP Echo-Request を *n* 回送信しても相手から有効な LCP Echo-Reply が受信されない場合、`pppd` は相手がダウンしているとみなします。このような場合、`pppd` は接続を切断します。このオプションを使用する場合は、`lcp-echo-interval` パラメータに 0 以外の値を指定する必要があります。このオプションを使用すると、ハードウェアモデム制御回線が使用できない状況で、モデムの切断などによって物理的な接続が切断されたあとに `pppd` を終了できます。

`lcp-echo-interval` *n*
このオプションが指定された場合、`pppd` は *n* 秒に 1 度、相手に LCP Echo-Request フレームを送信します。通常、相手は Echo-Reply を送信して Echo-Request に応答します。このオプションを `lcp-echo-failure` オプションとともに使用して、相手との接続が切断されたことを検出できます。

`lcp-max-configure` *n*
LCP Configure-Request の最大転送回数を *n* 回に設定します (デフォルト値は 10)。

`lcp-max-failure` *n*
LCP Configure-Reject の送信を開始するまでの LCP Configure-NAK の最大送信回数を *n* 回に設定します (デフォルト値は 10)。

`lcp-max-terminate` *n*
LCP Terminate-Request の最大送信回数を *n* 回に設定します (デフォルト値は 3)。

`lcp-restart` *n*
LCP 再開の間隔 (再送のタイムアウト) を *n* 秒に設定します (デフォルト値は 3)。

`linkname` *name*
リンクの論理名を *name* に設定します。`pppd` は、そのプロセス ID を含む `ppp-`*name*`.pid` というファイルを `/var/run` 内に作成します。これは、特定の相手に対するリンクをどの `pppd` インスタンスが処理しているかを特定するのに役立ちます。これは特権オプションです。

`local`
モデム制御回線を使用しません。このオプションを指定すると、`pppd` は、モデムからの CD (Carrier Detect) 信号の状態を無視し、DTR (Data Terminal Ready) 信号の状態を変更しません。

`logfd` *n*
ログメッセージをファイル記述子 *n* に送信します。`pppd` は、ログメッセージを syslog に送るとともに、最大 1 つのファイルまたはファイル記述子に送信するため、このオプションと `logfile` オプションを同時に指定することはできません。デフォルトでは、stdout でシリアルポートが開いている場合を除き、`pppd` はログメッセージを `stdout` (ファイル記述子 1) に送信します。

`logfile` *filename*
ログメッセージを syslog に送るとともに、ファイル *filename* に追加します。このファイルは追加モードで開かれ、`pppd` を起動したユーザーの特権が割り当てられます。

`login`
PAP を使用した相手の認証にシステムパスワードデータベースを使用し、ユーザーをシステムの `wtmp` ファイルに記録します。アクセスが許可されるためには、`/etc/ppp/pap-secrets` ファイルとシステムパスワードデータベースの両方に、相手のエントリが存在する必要があります。

`maxconnect` *n*
ネットワークトラフィックの使用開始から *n* 秒後に (つまり、最初のネットワーク制御プロトコルの開始から *n* 秒後に) 接続を切断します。最初の NCP の開始時と、残り時間が 5 分、2 分、および 0.5 分になったときに、LCP Time-Remaining メッセージが送信されます。

`maxfail` *n*
接続試行が *n* 回連続して失敗すると終了します。値 0 は制限を設定しないことを意味します。デフォルト値は 10 です。

`modem`
モデム制御回線を使用します。このオプションはデフォルトです。このオプションを指定すると、`pppd` は (接続スクリプトが指定されていなければ) モデムからの CD (Carrier Detect) 信号を待ってからシリアルデバイスを開き、接続の終了時および接続スクリプトの実行前に DTR (Data Terminal Ready) 信号を短時間落とします。

`ms-dns` <*addr*>
`pppd` が Microsoft Windows クライアントのサーバーとして動作している場合、このオプションを指定すると、`pppd` は 1 つまたは 2 つの DNS (Domain Name Server)

アドレスをクライアントに提供できます。このオプションの最初のインスタンスはプライマリ DNS アドレスを指定します。2 つ目のインスタンスがあれば、それはセカンダリ DNS アドレスを指定します。最初のインスタンスで指定された名前が複数の IP アドレスに解決される場合は、最初の 2 つのアドレスが使用されます。このオプションは、古いバージョンの `pppd` では `dns-addr` という名前でした。

`ms-lanman`
: `pppd` が Microsoft サーバーにクライアントとして接続し、認証に MS-CHAPv1 を使用する場合、このオプションを指定すると、Microsoft NT ではなく LAN Manager のパスワード形式が使用されます。

`ms-wins` <*addr*>
: `pppd` が Microsoft Windows クライアントまたは Samba クライアントのサーバーとして動作している場合、このオプションを指定すると、`pppd` は 1 つまたは 2 つの WINS (Windows Internet Name Services) サーバーアドレスをクライアントに提供できます。このオプションの最初のインスタンスはプライマリ WINS アドレスを指定します。2 つ目のインスタンスがあれば、それはセカンダリ WINS アドレスを指定します。`ms-dns` と同様に、指定された名前が複数の IP アドレスに解決される場合は、最初の 2 つのアドレスがプライマリおよびセカンダリとして使用されます。

`name` *name*
: 認証のために使用するローカルシステム名を *name* に設定します。これは特権オプションです。このオプションを指定すると、`pppd` は秘密情報ファイルの第 2 フィールドが *name* である行を使用して秘密情報を探し、相手の認証に使用します。さらに、`user` オプションで上書きされる場合を除き、ローカルシステムの認証を行うときに相手に送る名前として *name* を使用します。なお、`pppd` は *name* にドメイン名を付加しません。

`no-accm-test`
: LCP Echo-Request メッセージを使用した `asyncmap` (ACCM) チェックを無効にします。`lcp-echo-failure` が非同期回線に使用されている場合、`pppd` は最初の *n* 個の LCP Echo-Request メッセージにすべての制御文字を含めます。`asyncmap` が正しく設定されていない場合、リンクはランダムな障害で動作を継続するのではなく終了します。このオプションを指定すると、この機能は無効になります。

`noaccomp`
: HDLC Address/Control 圧縮を双方向 (送受信) で無効にします。

`noauth`
: 相手に自己認証を要求しません。このオプションは特権オプションです。

`nobsdcomp`
: BSD-Compresss 圧縮を無効にします。`pppd` は BSD-Compress 方式を使用したパケット圧縮を要求せず、同意もしません。`noccp` が指定されている場合、このオプションは不要です。

`noccp`
CCP (圧縮制御プロトコル) ネゴシエーションを無効にします。相手にバグがある場合や、`pppd` からの CCP ネゴシエーション要求で相手が混乱する場合にのみ、このオプションが必要です。CCP が無効になっている場合、BSD および deflate 圧縮を個別に無効にする必要はありません。

`nocrtscts`
シリアルポートのハードウェアフロー制御 (つまり RTS/CTS) を無効にします。`crtscts`、`nocrtscts`、`cdtrcts`、または `nocdtrcts` オプションが指定されていない場合、シリアルポートのハードウェアフロー制御の設定は変更されずにそのままになります。

`nocdtrcts`
このオプションの機能は `nocrtscts` と同じです。どちらのオプションも、両方の形式のハードウェアフロー制御を無効にします。

`nodefaultroute`
`defaultroute` オプションを無効にします。root 以外のユーザーが `pppd` でデフォルト経路を作成できないようにするには、このオプションを `/etc/ppp/options` ファイルに記述してください。

`nodeflate`
deflate 圧縮を無効にします。`pppd` は deflate 方式を使用したパケット圧縮を要求せず、同意もしません。`noccp` が指定されている場合、このオプションは不要です。

`nodeflatedraft`
インターネットドラフトの (誤って割り当てられた) deflate 圧縮のアルゴリズム番号を使用しません。`noccp` が指定されている場合、このオプションは不要です。

`nodetach`
制御端末から切り離しません。このオプションを指定しないと、標準入力上の端末以外のシリアルデバイスが指定された場合、`pppd` はフォークしてバックグラウンドプロセスになります。

`noendpoint`
マルチリンクエンドポイント判別子オプションを送信せず、受け入れることもありません。

`noident`
LCP 識別の使用を無効にします。LCP 識別メッセージが相手に送信されることはありませんが、受信されたメッセージはログに記録されます。LCP 識別を完全に無効にするには、このオプションを 2 回指定します。この場合、`pppd` は受信した LCP 識別メッセージへの応答として LCP Code-Reject を送信します。

`noip`
IPCP ネゴシエーションと IP 通信を無効にします。相手にバグがある場合や、`pppd` からの IPCP ネゴシエーション要求で相手が混乱する場合にのみ、このオプションを使用してください。

`noipv6`
IPv6CP ネゴシエーションと IPv6 通信を無効にします。IPv6 はデフォルトでは有効になっていません。

`noipdefault`
ローカル IP アドレスが指定されていない場合にデフォルトで行われる、ホスト名からローカル IP アドレスを (可能であれば) 決定する動作を無効にします。このオプションを指定した場合には、IPCP ネゴシエーション時に相手がローカル IP アドレスを指定する必要があります (明示的にコマンド行で指定されているか、オプションファイルで指定されている場合を除く)。

`nolog`
ファイルやファイル記述子にログメッセージを送りません。このオプションを指定すると、`logfd` オプションと `logfile` オプションが取り消されます。`nologfd` はこのオプションの別名として機能します。

`nomagic`
マジックナンバーネゴシエーションを無効にします。このオプションを指定すると、`pppd` はループバック回線を検出できません。相手にバグがある場合にのみ、このオプションを使用してください。「Serial line is looped back」というエラーメッセージを回避するためにこのオプションを使用しないでください。

`nopam`
この特権オプションは、プラグイン可能な認証モジュール (PAM) の使用を無効にします。このオプションを指定すると、`pppd` は標準の認証メカニズムに戻ります。デフォルトでは、PAM は使用されません。

`nopcomp`
プロトコルフィールド圧縮ネゴシエーションを送受信両方向で無効にします。

`nopersist`
接続が確立されたあとで切断されると、実行を終了します。`persist` または `demand` オプションが指定されていない場合、これはデフォルトです。

`noplink`
このオプションを指定すると、`pppd` は I_PLINK の代わりに I_LINK を使用します。これはデフォルトです。I_LINK を使用しているとき、システムは (SIGKILL が使用された場合でも) 終了したインタフェースをクリーンアップしますが、`ifconfig(1M)` による PPP ストリームの unplumb や、モジュールの動的な挿入、削除は許可しません。`ifconfig(1M)` の modinsert、modremove、または unplumb をサポートする必要がある場合は、`plink` オプションを使用してください。

`nopredictor1`
Predictor-1 圧縮を受け入れず、同意もしません。このオプションは互換性のために受け入れられます。実装では、Predictor-1 圧縮はサポートされていません。

`noproxyarp`
: `proxyarp` オプションを無効にします。ユーザーが `pppd` でプロキシ ARP エントリを作成できないようにするには、このオプションを `/etc/ppp/options` ファイルに記述してください。

`notty`
: 通常、`pppd` は端末デバイスを必要とします。このオプションを指定すると、`pppd` は擬似 tty のマスターとスレーブのペアを自分で割り当て、スレーブを端末デバイスとして使用します。`pppd` は、擬似 tty マスターとその標準入力および標準出力の間で文字を転送するために、文字分路として機能する子プロセスを作成します。このように、`pppd` は、その標準出力や標準入力が端末デバイスではない場合でも、標準出力で文字を送信し、標準入力で文字を受信します。このオプションを使用すると、すべての文字が文字分路プロセス経由で送受信される必要があるため、PPP インタフェースでのデータ転送の遅延と CPU オーバーヘッドが増加します。このオプションを使用する場合、デバイス名を明示的に指定することはできません。

`novj`
: Van Jacobson 形式の TCP/IP ヘッダー圧縮を送受信両方向で無効にします。

`novjccomp`
: Van Jacobson 形式の TCP/IP ヘッダー圧縮の接続 ID 圧縮を無効にします。このオプションを指定すると、`pppd` は Van Jacobson 形式で圧縮された TCP/IP ヘッダーから接続 ID を省略せず、相手にも省略を要求しません。`novj` が指定されている場合、このオプションは不要です。

`pam`
: この特権オプションは、PAM の使用を有効にします。このオプションを指定すると、`login` オプションと PAP 認証が使用されている場合、`pppd` は `pam(3PAM)` フレームワークを使用してサービス名「ppp」でユーザー認証を実行します。デフォルトでは、PAM は使用されません。

`papcrypt`
: このオプションを指定すると、`pppd` は、`/etc/ppp/pap-secrets` ファイル内の秘密情報と等しい暗号化前のパスワードを受け入れません。このオプションは、`pap-secrets` ファイル内の秘密情報が `crypt(3C)` 形式である場合に使用してください。

`pap-max-authreq` *n*
: PAP Authenticate-Request の最大転送回数を *n* 回に設定します (デフォルト値は 10)。

`pap-restart` *n*
: PAP 再開の間隔 (再送のタイムアウト) を *n* 秒に設定します (デフォルト値は 3)。

`pap-timeout` *n*
: 相手が PAP による自己認証を行うまで `pppd` が待機する最大時間を、*n* 秒に設定します (制限を設定しない場合は 0)。デフォルト値は 30 秒です。

`password` *string*
相手への認証に使用するパスワード文字列。

`persist`
接続が切断されたときに実行を終了するのではなく、再接続を試みます。

`plink`
このオプションを指定すると、`pppd` は I_LINK の代わりに I_PLINK を使用します。デフォルトでは I_LINK が使用されます。その場合、システムは (SIGKILL が使用された場合でも) 終了したインタフェースをクリーンアップしますが、`ifconfig(1M)` による PPP ストリームの unplumb や、モジュールの動的な挿入、削除は許可しません。このオプションは、`ifconfig(1M)` の modinsert、modremove、または unplumb をサポートする必要がある場合に使用してください。`plumbed` オプションも参照してください。

`plugin` *filename*
共有ライブラリオブジェクトファイル *filename* をプラグインとして読み込みます。これは特権オプションです。filename でパスが明示的に指定されている場合を除き、読み込むオブジェクトは `/etc/ppp/plugins` と `/usr/lib/inet/ppp` で (この順に) 検索されます。

`plumbed`
このオプションを指定すると、`pppd` は plumb されたインタフェースを探してセッションに使用します。IPv4 アドレス、IPv6 インタフェース ID、リンク MTU は、別の方法で指定されていない場合、選択されたインタフェースからコピーされます。このモードは、以前の `aspppd` 実装に似た機能を持っているため、`ifconfig(1M)` を使用する外部アプリケーションとともに `pppd` を使用する場合に役立つことがあります。

`pppmux` *timer*
PPP 多重化オプションのネゴシエーションを有効にし、転送多重化のタイムアウトを *timer* ミリ秒に設定します。

`privgroup` *group-name*
グループ *group-name* のメンバーに特権オプションの使用を許可します。これは特権オプションです。 *group-name* のメンバーが `pppd` を使用して root になる可能性も排除できないため、このオプションを使用するときは注意が必要です。これは *group-name* のメンバーを `root` または `sys` グループに入れることに等しいとみなしてください。

`proxyarp`
このシステムのアドレス解決プロトコル (ARP) テーブルに、相手の IP アドレスとこのシステムの Ethernet アドレスのエントリを追加します。このオプションを使用すると、ほかのシステムには、相手がローカルの Ethernet 上にあるように見えます。PPP リンク上のリモートアドレスは、Ethernet インタフェースに割り当てられているものと同じサブネットに属している必要があります。

`pty` *script*
特定の端末デバイスではなく、コマンド *script* をシリアル通信に使用することを指定します。`pppd` は擬似 tty のマスターとスレーブのペアを自分で割り当て、スレーブを端末デバイスとして使用します。*script* は、擬似 tty マスターを標準入力および標準出力として子プロセス内で実行されます。このオプションを使用する場合、デバイス名を明示的に指定することはできません。(注: `record` オプションを `pty` オプションとともに使用すると、子プロセスの標準入力および標準出力にパイプが適用される。)

`receive-all`
このオプションを指定すると、`pppd` は、相手からの制御文字をすべて、受信 `asyncmap` でマークされているものも含めて受け入れます。このオプションを指定しないと、`pppd` は RFC 1662 で指定されている文字を破棄します。ダイアルバックの実装でよく見受けられるように、相手にバグがある場合にのみ、このオプションを使用するようにしてください。

`record` *filename*
このオプションを指定すると、`pppd` は送受信したすべての文字を *filename* というファイルに記録します。*filename* は、ユーザーのユーザー ID とアクセス権を使用して、追加モードで開かれます。このオプションでは、擬似 tty とプロセスを使用して、その擬似 tty と実際のシリアルデバイスの間で文字を転送するため、PPP インタフェースでのデータ転送の遅延と CPU オーバーヘッドが増加します。文字はタグ付き形式でタイムスタンプとともに記録され、`pppdump`(1M) プログラムを使用して可読形式で表示できます。通常、このオプションは `pppd` のカーネル部分 (特に CCP 圧縮アルゴリズム) のデバッグに使用され、接続構成の問題のデバッグには使用されません。`debug` オプションを参照してください。

`remotename` *name*
認証のために使用する仮定のリモートシステム名を *name* に設定します。Microsoft WindowsNT は CHAP チャレンジメッセージ内にシステム名を指定しないため、その問題を回避するためにこのオプションがよく使用されます。

`refuse-chap`
このオプションを指定すると、`pppd` は、標準のチャレンジハンドシェーク認証プロトコル (CHAP) を使用して相手に対する自己認証を行うことに同意しません。(MS-CHAP には影響しない。)

`refuse-mschap`
MS-CHAPv1 を使用して相手に対する自己認証を行うことに同意しません。このオプションが指定されている場合、相手からの MS-CHAPv1 認証の要求は LCP Configure-Nak で拒否されます。このオプションを指定しても、CHAP のほかの形式が無効になることはありません。

`refuse-mschapv2`
MS-CHAPv2 を使用して相手に対する自己認証を行うことに同意しません。このオプションが指定されている場合、相手からの MS-CHAPv2 認証の要求は LCP Configure-Nak で拒否されます。このオプションを指定しても、CHAP のほかの形式が無効になることはありません。

`refuse-pap`
このオプションを指定すると、`pppd` は、パスワード認証プロトコル (PAP) を使用して相手に対する自己認証を行うことに同意しません。

`require-chap`
標準の CHAP 認証を使用して自己認証を行うことを相手に要求します。MS-CHAP には影響しません。

`require-mschap`
MS-CHAPv1 認証を使用して自己認証を行うことを相手に要求します。

`require-mschapv2`
MS-CHAPv2 認証を使用して自己認証を行うことを相手に要求します。

`require-pap`
PAP 認証を使用して自己認証を行うことを相手に要求します。

`show-password`
このオプションを指定すると、`pppd` は PAP パケットの内容をログに記録するときに、パスワード文字列をログメッセージに含めます。

`silent`
このオプションを指定すると、`pppd` は相手から有効な LCP パケットを受信するまで、接続を開始するための LCP パケットを送信しません。これは旧バージョンの `pppd` の passive オプションと同様のオプションで、互換性のために残されていますが、現在の `passive` オプションをお勧めします。

`small-accm-test`
デフォルトでは、`asyncmap` (ACCM) の設定をチェックするとき、`pppd` は使用可能な 256 個の値をすべて使用します。`no-accm-test` を参照してください。このオプションを指定すると、標準の ACCM ネゴシエーションの影響を受ける 32 個の値だけがテストに使用されます。このオプションは、非常に低速のリンクで役立ちます。

`socket` *host*:*port*
TCP を使用して指定のホストとポートに接続し、この接続上で PPP を実行します。

`sync`
非同期ではなく同期 HDLC シリアル符号化を使用します。このオプションを指定する場合、`pppd` で使用されるデバイスに同期のサポートが組み込まれている必要があります。現在は、`zs`、`se`、および `hsi` ドライバがサポートされています。

`unit` *n*
可能であれば、PPP インタフェースのユニット番号を *n* に設定します。

`updetach`
このオプションを指定すると、`pppd` は PPP 接続を確立したあと、制御端末から切り離します。このオプションが指定されている場合、接続スクリプト (通常は

chat(1M)) によって stderr に送信されるメッセージと、debug オプションからのデバッグメッセージは、pppd の標準出力に送られます。

usehostname
認証のために使用するローカルシステム名として、ホスト名を (ドメイン名が指定されている場合はドメイン名を付加して) 使用することを強制します。このオプションは name オプションに優先します。name オプションは特権オプションなので、通常このオプションは不要です。

usepeerdns
最大 2 つの DNS サーバーアドレスを相手に要求します。相手からアドレスが提供された場合、それらは環境変数 DNS1 および DNS2 で /etc/ppp/ip-up スクリプトに渡されます。また、pppd は相手から提供されたアドレスを使用して、1 つまたは 2 つの nameserver 行を含む /etc/ppp/resolv.conf ファイルを作成します。

user *name*
相手に対してローカルシステムの認証を行うときに使用する名前を *name* に設定します。

vj-max-slots *n*
Van Jacobson の TCP/IP ヘッダー圧縮/圧縮解除コードで使用する接続スロット数を *n* に設定します。2 以上かつ 16 以下の値を指定する必要があります。

welcome *script*
PPP ネゴシエーション開始前かつ、(もしあれば) 接続スクリプトの完了後に、*script* で指定された実行可能コマンドまたはシェルコマンドを実行します。このオプションの値が特権ソースから読み込まれた場合、非特権ユーザーがそれを上書きすることはできません。

xonxoff
シリアルポートのデータフロー制御にソフトウェアフロー制御 (XON/XOFF) を使用します。

廃止されたオプション

次のオプションは廃止されました。

+ua *name*
PAP ユーザー名とパスワードをファイル *name* から読み込みます。このファイルには、名前とパスワードを表す 2 行が含まれている必要があります。相手が PAP 認証を要求したときに、この名前とパスワードが相手に送信されます。

+ipv6
インタフェース識別子を指定せずに IPv6 と IPv6CP を有効にします。

--version
バージョン番号を表示して終了します。

--help
短いヘルプメッセージを表示して終了します。

拡張機能説明

ここでは、pppd のその他の機能について説明します。

セキュリティー

システム管理者は pppd を使用して、サーバー自体やそのサーバーが稼働しているネットワークのセキュリティーを損なうことなく、正当なユーザーにサーバーマシンへの PPP アクセスを提供できます。相手の認証済み識別情報 (存在する場合) に基づいて相手が使用できる IP アドレスを制限し、また、非特権ユーザーが使用できるオプションを制限することで、アクセス制御が提供されます。潜在的に安全でない設定を許可するオプションは、特権オプションです。特権オプションが受け入れられるのは、システム管理者の制御下にあるファイル内でのみ、または pppd が root によって実行されている場合のみです。

デフォルトでは、pppd は、特定の IP アドレスへの経路がシステムに設定されていない場合に限り、認証されていない相手にその IP アドレスの使用を許可します。たとえば、より広いインターネットへの常時接続を備えたシステムの場合、通常はデフォルト経路が設定されているため、接続を確立するにはどの相手も自己認証を行う必要があります。そのようなシステムでは、auth オプションはデフォルトです。逆に、インターネットへの接続を 1 つだけ含む PPP リンクを備えたシステムでは、デフォルト経路が設定されていない可能性が高いため、相手は自己認証を行うことなく、実質的に任意の IP アドレスを使用できます。

セキュリティーに影響するオプションは特権オプションです。非特権ユーザーが pppd を実行する場合、コマンド行でも、ユーザーの $HOME/.ppprc ファイルでも、または file オプションを使用して読み込まれるオプションファイルでも、特権オプションを使用することはできません。特権オプションは、/etc/ppp/options ファイル内、または call オプションを使用して読み込まれるオプションファイル内で使用できます。root ユーザーが pppd を実行する場合、特権オプションは制限なく使用できます。/etc/ppp/options ファイルが存在しない場合は、root だけが pppd を起動できます。root 以外の通常ユーザーが pppd にアクセスするには、/etc/ppp/options ファイルを作成する必要があります (ただし、空でもよい)。

pppd はデバイスを開くとき、pppd を起動したユーザーのユーザー ID または root UID (つまり、0) を使用します。これは、デバイス名を指定したのがユーザーかシステム管理者かによって異なります。デバイス名が特権ソース (つまり、/etc/ppp/options または call オプションで読み込まれるオプションファイル) から読み込まれた場合、pppd は完全な root 特権を使用してデバイスを開きます。したがって、システム管理者が適切なファイルを /etc/ppp/peers に作成することで、ユーザーは通常であればアクセスできないデバイスを使用して PPP 接続を確立できるようになります。それ以外の場合、pppd は、pppd を起動したユーザーの実 UID を使用してデバイスを開きます。

認証

認証処理では、一方が他方に自分の名前と何らかの秘密情報を送ることにより、自分の身元を認めさせます。認証では、最初の一方が「クライアント」になり、他方が「サーバー」になります。認証用の名前にはそれぞれのインターネットホスト名を使用できますが、これは必須ではありません。

pppd は 4 つの認証プロトコルをサポートしています。つまり、パスワード認証プロトコル (PAP) と、3 つの形式のチャレンジハンドシェーク認証プロトコル (CHAP) で

す。PAP プロトコルの場合、クライアントは自己認証のために、自己の名前と平文パスワードをサーバーに送ります。CHAP の場合は、サーバーがチャレンジをクライアントに送ることで認証交換を開始します。これに対してクライアントは、自己の名前、および共有シークレットとチャレンジから得られるハッシュ値で応答する必要があります。

PPP プロトコルは対称的です。つまり、両者とも、他方に対する自己認証を要求される可能性があります。交換ごとに異なる認証プロトコルと名前を使用できます。

デフォルトでは、`pppd` は、要求された場合に自己認証を行い、相手に自己認証を要求しません。ただし、特定のプロトコルでは、自己認証を行うための秘密情報がない場合には `pppd` は自己認証を行いません。

`pppd` は認証用の秘密情報を `/etc/ppp/pap-secrets` ファイル (PAP) および `/etc/ppp/chap-secrets` ファイル (CHAP) に保存します。どちらのファイルも同じ形式を使用します。`pppd` は、ほかのシステムに対する自己認証、および自己に対するほかのシステムの認証に、秘密情報ファイルを使用します。

秘密情報ファイルでは、1 行に 1 つの秘密情報が格納されます。秘密情報は、クライアントとサーバーの特定の組み合わせに固有で、そのクライアントがそのサーバーに対して自己認証を行う場合にのみ使用できます。秘密情報ファイルの各行には、少なくとも 3 つのフィールドがあり、クライアント名、サーバー名、およびそのあとに秘密情報が格納されます。多くの場合、これらの 3 つのフィールドに続き、クライアントがサーバーに接続するために使用する IP アドレスが記述されます。

秘密情報ファイルは、クライアント名、サーバー名、および秘密情報が各フィールドに 1 つの文字列で記述されたものとして構文解析されます。1 つの文字列にスペース文字や特殊文字が含まれる場合は、クォートまたはエスケープする必要があります。3 つのフィールドのいずれでも、大文字と小文字は区別されます。

秘密情報がアットマーク (@) で始まっている場合は、秘密情報が含まれているファイルの名前がそのあとに続きます。クライアント名またはサーバー名が単一のアスタリスク (*) で表されている場合は、任意の名前に一致します。一致する対象を選択するとき、`pppd` はワイルドカードが最小となるものを選択します。`pppd` は、同じ行の残りの文字列を、そのクライアントに許可される IP アドレスと解釈します。3 つの文字列しかない行や、最初の文字列がハイフン (-) で始まっている行には、IP アドレスを記述できません。任意のアドレスを許可するには、アスタリスク (*) を使用します。感嘆符 (!) で始まっているアドレスは、そのアドレスが受け入れられないことを示します。アドレスの後ろにスラッシュ (/) と数値 *n* が付加されている場合は、サブネット全体 (最上位 *n* ビットに同じ値を持つすべてのアドレス) を示します。この形式で、アドレスの後ろにプラス記号 (+) が付加されている場合は、使用中の PPP ネットワークインタフェースのユニット番号に基づいて、サブネットの 1 つのアドレスが承認されることを示します。この場合、アドレスのホスト部分は、そのユニット番号に 1 を加えた値に設定されます。

相手を認証するとき、pppd は秘密情報ファイルで、最初のフィールドに相手の名前があり、2 番目のフィールドにローカルシステムの名前がある秘密情報を選択します。ローカルシステム名のデフォルトはホスト名であり、domain オプションが使用されている場合はドメイン名が付加されます。usehostname オプションが使用されている場合を除き、name オプションでこのデフォルトを上書きできます。

相手に対して自己認証を行うとき、pppd はまず、相手に対して自己の識別情報として使用する名前を決定します。この名前は user オプションで指定されます。user オプションが使用されていない場合、名前はデフォルトでローカルシステムのホスト名になります。次に、pppd は秘密情報ファイルで、最初のフィールドにローカルシステムの名前があり、2 番目のフィールドに相手の名前がある秘密情報を選択します。標準の CHAP 認証が使用される場合、相手の名前はチャレンジパケット内で送信されているため、pppd には相手の名前がわかります。しかし、MS-CHAP または PAP が使用される場合、pppd はユーザーが指定したオプションから相手の名前を決定する必要があります。ユーザーは相手の名前を直接 remotename オプションで指定できます。それ以外の場合、リモート IP アドレスが数値形式でなく名前で指定されたときは、その名前を相手の名前として使用します。これに失敗すると、pppd は相手の名前として NULL 文字列を使用します。

相手を PAP で認証するとき、提供されたパスワードは秘密情報ファイルのデータと比較されます。パスワードが秘密情報と一致しない場合、パスワードは crypt() を使用して暗号化され、再度秘密情報と比較されます。papcrypt オプションが指定されている場合、暗号化されていない最初の比較はセキュリティー向上のために省略されるため、エントリは暗号化された crypt(3C) 形式でなければなりません。

login オプションが指定されている場合は、ユーザー名とパスワードもシステムパスワードデータベースと照合されます。したがって、pap-secrets ファイルを設定することで、特定のユーザーだけに PPP アクセスを許可したり、ユーザーに使用を許可する IP アドレスのセットを制限したりできます。通常、login オプションを使用する場合、/etc/ppp/pap-secrets 内の秘密情報を "" とすることで、相手が提供する任意のパスワードに一致するようになります。これにより、同じ秘密情報を 2 か所で保持する必要がなくなります。login が使用されている場合に pam オプションを指定すると、pam(3PAM) を介したアクセス制御が有効になります。

IPCP (またはほかのネットワークプロトコル) を開始するには、その前に認証が完了している必要があります。相手が自己認証を要求されている場合にそれが失敗すると、pppd は LCP を閉じ、リンクを切断します。IPCP ネゴシエーションで得られたリモートホストの IP アドレスが受け入れられないものだった場合、IPCP は閉じられます。IP パケットは IPCP が開いているときだけ送受信されます。

自己認証できないホストに接続を許可し、制限された IP アドレスの 1 つの使用を許可するには、クライアント名と秘密情報に空の文字列を指定した 1 行を pap-secrets ファイルに追加します。

特定の相手に関する追加の pppd オプションは、秘密情報エントリの後ろに 2 つのダッシュ (--) で区切って指定できます。次に例を示します。

```
peername servername secret ip-address -- novj
```

ルーティング

IPCP ネゴシエーションが完了すると、pppd はカーネルに、PPP インタフェースで使用するローカル IP アドレスおよびリモート IP アドレスを通知します。また、リンクのリモート側へのホスト経路を作成して、IP パケットを交換できるようにします。ほかのマシンとの通信には、通常、ルーティングテーブルやアドレス解決プロトコル (ARP) テーブルをさらに変更する必要があります。ほとんどの場合、これには defaultroute や proxyarp オプションで十分ですが、さらに操作が必要になることもあります。さらに操作が必要な場合は、/etc/ppp/ip-up スクリプトまたは経路指定デーモンを使用してください。

リモートホスト経由のデフォルト経路を追加するには、defaultroute オプションを使用します。通常、このオプションは「クライアント」システム、つまり、PPP リンクを使用して一般的なインターネットにアクセスするエンドノードに使用されます。

たとえばサーバーマシンが LAN に接続されている場合など、ほかのホストがリモートホストと通信できるようにするために、プロキシ ARP の使用が望ましい場合もあります。proxyarp オプションを指定すると、pppd はリモートホストと同じサブネット上にあるネットワークインタフェースを探します。つまり、ブロードキャストと ARP をサポートし、ポイントツーポイントインタフェースやループバックインタフェースでない、現在動作中のインタフェースを探します。そのようなインタフェースが見つかった場合、pppd は、リモートホストの IP アドレスとその見つかったネットワークインタフェースのハードウェアアドレスを使用して、公開された永続的な ARP エントリを作成します。

demand オプションが使用されている場合、IPCP 開始時にインタフェースの IP アドレスは設定済みです。pppd は、インタフェースの設定に使用したものと同じアドレスをネゴシエーションで取得できない場合、インタフェースの IP アドレスをネゴシエーションで取得されたアドレスに変更します。これにより、既存の接続が中断される場合があります。動的 IP アドレス割り当てを実行する相手に要求時ダイアルを使用することはお勧めしません。

スクリプト

pppd は処理のさまざまな段階においてスクリプトを起動し、サイト固有の補助的な処理を実行します。これらのスクリプトは、シェルスクリプトまたは実行可能プログラムです。pppd はスクリプトが終了するまで待機しません。スクリプトは root として (実ユーザー ID および実効ユーザー ID を 0 に設定して) 実行されるため、ルーティングテーブルの更新や特権デーモンの実行などのタスクを実行できます。これらのスクリプトの内容によってシステムセキュリティーが損なわれないよう注意してください。pppd は標準入力、標準出力、および標準エラーを /dev/null にリダイレクトし、リンクの情報を指定するいくつかの環境変数を除いて環境変数を空にして、スクリプトを実行します。pppd の環境変数は次のとおりです。

DEVICE　シリアル tty デバイスの名前。

IFNAME　ネットワークインタフェースの名前。

IPLOCAL　リンクのローカル側の IP アドレス。IPCP が開始したときにのみ設定されます。

IPREMOTE　リンクのリモート側の IP アドレス。IPCP が開始したときにのみ設定されます。

PEERNAME　相手の認証された名前。相手が自己認証を行なった場合にのみ設定されます。

SPEED　tty デバイスのボーレート。

ORIG_UID　pppd を起動したユーザーの実ユーザー ID。

PPPLOGNAME　pppd を起動した実ユーザー ID のユーザー名。これは常に設定されます。

pppd は ip-down および auth-down スクリプト用に次の変数も設定します。

CONNECT_TIME　PPP ネゴシエーションの開始から接続切断までの秒数。

BYTES_SENT　接続中にシリアルポートのレベルで送信されたバイト数。

BYTES_RCVD　接続中にシリアルポートのレベルで受信されたバイト数。

LINKNAME　linkname オプションで設定された、リンクの論理名。

存在する場合、pppd は次のスクリプトを呼び出します。存在しなくてもエラーではありません。

/etc/ppp/auth-up　リモートシステムの自己認証が正常に行われたあとに実行されるプログラムまたはスクリプト。5 つのコマンド行引数 interface-name peer-name user-name tty-device speed を使用して実行されます。たとえば、noauth オプションが使用されているときなど、相手が自己認証を行わない場合、このスクリプトは実行されません。

/etc/ppp/auth-down　あらかじめ /etc/ppp/auth-up が実行された場合で、リンクがダウンしたときに実行されるプログラムまたはスクリプト。/etc/ppp/auth-up と同じ方法で、同じパラメータを使用して実行されます。

/etc/ppp/ip-up　リンクが IP パケットの送受信に使用可能になった (つまり、IPCP が完了した) ときに実行されるプログラムまたはスクリプト。6 つのコマンド行引数 interface-name tty-device speed local-IP-address remote-IP-address ipparam を使用して実行されます。

/etc/ppp/ip-down　リンクが IP パケットの送受信に使用不可になったときに実行されるプログラムまたはスクリプト。このスクリプトは、/etc/ppp/ip-up スクリプトの影響を元に戻すために使用できます。ip-up スクリプトと同じ方法で、同じパラメータを使用して実行されます。

/etc/ppp/ipv6-up　/etc/ppp/ip-up と同様ですが、リンクが IPv6 パケットの送受信に使用可能になったときに実行されます。6 つのコマンド行引数 interface-name tty-device speed local-link-local-address remote-link-local-address ipparam を使用して実行されます。

/etc/ppp/ipv6-down　/etc/ppp/ip-down と同様ですが、リンクが IPv6 パケットの送受信に使用不可になったときに実行されます。ipv6-up スクリプトと同じパラメータを使用して実行されます。

使用例

例 1　auth オプションを使用する

次の例では、/etc/ppp/options ファイルに auth オプションが含まれているものとします。

pppd は一般に ISP へのダイアルアウトに使用されます。これは pppd call isp コマンドを使用して実行できます。/etc/ppp/peers/isp ファイルには次のような行が含まれています。

```
cua/a 19200 crtscts connect '/usr/bin/chat -f /etc/ppp/chat-isp' noauth
```

この例では、chat(1M) を使用して ISP のモデムにダイアルし、必要なログインシーケンスを処理します。/etc/ppp/chat-isp ファイルは chat によって使用され、たとえば次のような内容を含んでいます。

```
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
ABORT "ERROR"
ABORT "NO ANSWER"
ABORT "BUSY"
ABORT "Username/Password Incorrect"
"" "at"
OK "at&f&d2&c1"
OK "atdt2468135"
"name:" "^Umyuserid"
"word:" "\qmypassword"
"ispts" "\q^Uppp"
"~-^Uppp-~"
```

chat スクリプトの詳細については、chat(1M) のマニュアルページを参照してください。

例 2　pppd を proxyarp とともに使用する

pppd では、ダイアルイン PPP サービスをユーザーに提供することもできます。ユーザーがすでにログインアカウントを持っている場合、PPP サービスを設定するもっとも簡単な方法は、ユーザーが自分のアカウントにログインして pppd を次の例のように実行することです。

```
example% pppd proxyarp
```

例 3　ユーザーが PPP 機能を使用できるようにする

ユーザーが PPP 機能を使用できるようにするには、そのユーザーのマシンに IP アドレスを割り当て、/etc/ppp/pap-secrets または /etc/ppp/chap-secrets にエントリを作成します。これにより、ユーザーのマシンは自己認証を行うことができます。たとえば、joespc というマシンを使用しているユーザー Joe が server というマシンにダイアルインし、joespc.my.net という IP アドレスを使用できるようにするには、次のエントリを /etc/ppp/pap-secrets または /etc/ppp/chap-secrets ファイルに追加します。

```
joespc    server    "joe's secret"    joespc.my.net
```

または、たとえば ppp という別のユーザー名を作成し、そのログインシェルを /usr/bin/pppd、ホームディレクトリを /etc/ppp とする方法もあります。この方法で pppd を実行する場合、オプションは /etc/ppp/.ppprc ファイルに記述します。

複雑なシリアル接続を使用している場合は、asyncmap a0000 を使用して、XON (^Q) や XOFF (^S) などの制御文字をエスケープすると役立つ場合があります。パスに telnet が含まれている場合は、^] をエスケープしてください (asyncmap 200a0000)。パスに rlogin コマンドが含まれている場合は、オプションに escape ff オプションを追加してください。これは、rlogin により、0xff、0xff、0x73、0x73、およびそれに続く 8 バイトで構成されるウィンドウサイズ変更シーケンスがストリームから削除されるためです。

終了ステータス

pppd の終了ステータスは、エラーやリンク切断理由を示します。終了ステータスの値は次のとおりです。

0　pppd は切り離されたか、接続が正常に確立されたあとで相手の要求で切断されました。

1　即時の致命的なエラーが発生しました。たとえば、重要なシステムコールが失敗しました。

2　指定されたオプションにエラーが検出されました。たとえば、同時に指定することのできないオプションが指定されている場合や、/etc/ppp/options が存在せずユーザーが root でない場合などです。

3　pppd が setuid-root でなく、起動したユーザーが root ではありません。

4 カーネルが PPP をサポートしていません。たとえば、PPP カーネルドライバが含まれていないか読み込み不可能です。

5 pppd は SIGINT、SIGTERM、または SIGHUP 信号が送信されたため終了しました。

6 シリアルポートをロックできませんでした。

7 シリアルポートを開くことができませんでした。

8 接続スクリプトが失敗し、ゼロ以外の終了ステータスを返しました。

9 pty オプションの引数として指定されたコマンドを実行できませんでした。

10 ネットワークプロトコルを実行できなかったため、PPP ネゴシエーションが失敗しました。

11 相手システムが自己認証に失敗したか、自己認証を拒否しました。

12 リンクは正常に確立されましたが、アイドルであったために切断されました。

13 リンクは正常に確立されましたが、接続時間制限に達したために切断されました。

14 コールバックのネゴシエーションが行われたため、まもなく着呼があります。

15 相手がエコー要求に応答しないためリンクが切断されました。

16 モデムの切断によってリンクが切断されました。

17 シリアルループバックが検出されたため PPP ネゴシエーションが失敗しました。

18 ゼロ以外の終了ステータスが返されたため、init スクリプトが失敗しました。

19 相手への認証が失敗しました。

ファイル

/var/run/spppn.pid	PPP インタフェースユニット *n* に対応する pppd プロセスのプロセス ID を格納します。
/var/run/ppp-*name*.pid	論理リンク名に対応する pppd プロセスのプロセス ID を格納します (linkname オプションを参照)。
/etc/ppp/pap-secrets	PAP 認証で使用するユーザー名、パスワード、および IP アドレスを格納します。このファイルは root が所有し、ほかのユーザーには読み書きできないようにする必要があります。そうでない場合、pppd は警告をログに記録します。

/etc/ppp/chap-secrets	すべての形式の CHAP 認証で使用する名前、秘密情報、および IP アドレスを格納します。/etc/ppp/pap-secrets ファイルは root が所有し、ほかのユーザーには読み書きできないようにする必要があります。そうでない場合、pppd は警告をログに記録します。
/etc/ppp/options	pppd のシステムデフォルトオプションを記述します。ユーザーデフォルトオプションやコマンド行オプションより前に読み込まれます。
$HOME/.ppprc	ユーザーのデフォルトオプションを記述します。/etc/ppp/options.*ttyname* より前に読み込まれます。
/etc/ppp/options.*ttyname*	使用中のシリアルポートのシステムデフォルトオプションを記述します。$HOME/.ppprc のあとで読み込まれます。このファイル名の *ttyname* 部分を構築するために、ポート名から先頭の /dev/ が取り除かれ (存在する場合)、スラッシュ文字 (/) があればドットに変換されます。
/etc/ppp/peers	このディレクトリのオプションファイルには、pppd が root 以外のユーザーによって起動された場合でも使用できる特権オプションを指定します。システム管理者がこのディレクトリにオプションファイルを作成することにより、非特権ユーザーは、信頼できる特定の相手に限り、相手に認証を要求せずにダイアルアウトできるようになります。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/network/ppp
インタフェースの安定性	確実

関連項目

chat(1M), ifconfig(1M), crypt(3C), pam(3PAM), attributes(5)

『RFC 2472 – IP Version 6 Over PPP』、Haskin, D. および Allen, E. 著、Network Working Group 発行、1998 年 12 月

『RFC 1144, Compressing TCP/IP Headers for Low-Speed Serial Links』、Jacobson, V. 著、Network Working Group 発行、1990 年 2 月

『RFC 1334, PPP Authentication Protocols』、Lloyd, B. および Simpson, W. 著、Network Working Group 発行、1992 年 10 月

『RFC 1332, The PPP Internet Protocol Control Protocol (IPCP)』、McGregor, G. 著、Network Working Group 発行、1992 年 5 月

『RFC 1321, The MD5 Message-Digest Algorithm』、Rivest, R. 著、Network Working Group 発行、1992 年 4 月

『RFC 1661, The Point-to-Point Protocol (PPP)』、Simpson, W. 著、Network Working Group 発行、1994 年 7 月

『RFC 1662, HDLC-like Framing』、Simpson, W. 著、Network Working Group 発行、1994 年 7 月

注意事項

次の信号は pppd の動作に影響を与えます。

SIGINT、SIGTERM
: リンクを切断し、シリアルデバイスの設定を復元してから、実行を終了します。

SIGHUP
: リンクを切断し、シリアルデバイスの設定を復元し、シリアルデバイスを閉じます。persist または demand オプションが指定されている場合、pppd はシリアルデバイスを再度開き、抑止期間を置いてから新しい接続を開始しようとします。それ以外の場合、pppd は終了します。抑止期間中に SIGHUP を受信すると、pppd は抑止期間をすぐに終了します。

SIGUSR1
: debug オプションのステータスを切り替え、リンクのステータス情報をログに出力します。

SIGUSR2
: この信号を受信すると、pppd は圧縮のネゴシエーションを再度行います。これは、致命的な圧縮解除エラーの結果として圧縮が無効になったあとで、再度圧縮を有効にする場合に役立ちます。通常、致命的な圧縮解除エラーは実装のバグを示します。

診断

メッセージは LOG_DAEMON 機能を使用して syslog デーモンに送られます。エラーメッセージやデバッグメッセージを確認するには、/etc/syslog.conf ファイルを編集してメッセージの送り先となるデバイスやファイルを指定するか、updetach または logfile オプションを使用します。

debug オプションを指定すると、送受信されたすべての LCP、PAP、CHAP、または IPCP 制御パケットの内容がログに記録されます。これは、PPP ネゴシエーションが成功しない場合や、認証が失敗する場合に役立ちます。

SIGUSR1 信号を送信することによっても、デバッグを有効にしたり無効にしたりできます。この信号は、pppd プロセスに対するトグルとして機能します。

名前

pppoec – PPPoE chat utility

形式

```
pppoec [-omillisecs] [-smillisecs] [-v] device
     [service [ [except]server... [only]]]

pppoec [-omillisecs] [-v] -i [device]
```

機能説明

The `pppoec` utility implements the client-side negotiation of PPPoE. It is intended to be used with the `pppd(1M)` `connect` option, in the same manner as the `chat(1M)` utility is used for asynchronous dial-up PPP.

When given with the `-i` flag, `pppoec` sends out a broadcast query on the given interface named by the *device* parameter. You can specify no other arguments in this mode. All responding PPPoE servers and the offered services are displayed on standard output.

Otherwise, when given without the `-i` flag, `pppoec` does the full PPPoE client-side negotiation. The *device* parameter is the intended Ethernet interface, and must already be plumbed with `sppptun(1M)`. The optional *service* parameter specifies a particular service desired; other offered services will be ignored. The optional *server* parameter specifies a specific server desired. You can specify *server* as an Ethernet address in the usual x:x:x:x:x:x format (with "*" in any of the six byte positions interpreted to mean "any"), or as a symbolic name resolved through `/etc/ethers` (or NIS), or as a PPPoE access concentrator name. The sense of the match (true or false) can be inverted by specifying the keyword `except` before this string. This parameter can be specified more than once, and the first match is taken.

If you specify the *server* parameter, then the selected servers become "preferred." If no preferred server responds, then the first responding server is used instead. To exclude non-matching servers entirely, append the keyword `only`.

オプション

The following options are supported:

`-i` Sends out broadcast query over interface specified by *device*.

`-o` Sets the initial wait time in milliseconds for PADO from the server before PADI is retried. The default is 500 milliseconds for normal operation, or 3000 milliseconds (3 seconds) for inquiry (`-i`) mode.

`-s` Sets the initial wait time in milliseconds for PADS from the server before PADR is retried. The default is 2000 milliseconds (2 seconds).

`-v` Displays verbose progress messages, including all PPPoE messages sent, and all state machine transitions.

You normally do not need to adjust the parameters set with `-o` and `-s`. They are provided for coping with unusually slow servers.

オペランド

The following operands are supported:

device plumbed Ethernet interface

server preferred server or, if you specify `only`, the specified server

service desired service; other available services are ignored

使用例

例 1 Connecting to Any Service on hme0

The following command enables you to connect to any PPPoE service on `hme0`:

```
# /usr/bin/pppd sppptun plugin pppoe.so \
connect "/usr/lib/inet/pppoec hme0" debug
```

Often, a command such as the preceding is specified in an `/etc/ppp/peers` file instead. For example, enter the following in `/etc/ppp/peers/myisp`:

```
sppptun
plugin pppoe.so
connect "/usr/lib/inet/pppoec hme0"
debug
```

To invoke the PPP connection described in the file, enter:

```
% /usr/bin/pppd call myisp
```

Note that, because the `/etc/ppp/peers` files are considered privileged by `pppd`, you need not be root to invoke the preceding command.

例 2 Connecting to a Particular Service

A more complex example: on `hme0`, connect to only the `internet` service offered by PPPoE servers with access concentrator name `isp`, but not to any Ethernet addresses starting with `40:0:1a`.

```
# /usr/lib/inet/pppoec hme0 internet except 40:0:1a:*:*:* isp only
```

Note that the `except 40:0:1a:*:*:*` filter must come before `isp`, because the filters are first-match.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/usr/lib/inet/pppoec`	executable command
`/dev/sppptun`	Solaris PPP tunneling device driver.
`/etc/ppp/connect-errors`	usual location of error output (see DIAGNOSTICS, below)

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ppp/tunnel

関連項目

pppd(1M), sppptun(1M), pppoed(1M), sppptun(7M)

RFC 2516, Method for Transmitting PPP Over Ethernet (PPPoE), Mamakos et al, February 1999

診断

Error messages are written to standard error, which is normally redirected by pppd to /etc/ppp/connect-errors. The errors can also be redirected to pppd's standard output by using the updetach option.

If you specify the -v, verbose progress messages are displayed, including all PPPoE messages sent, and all state machine transitions. Specifying the updetach or nodetach pppd option is helpful when using verbose mode.

名前

pppoed – PPPoE server daemon

形式

`ppoed [options]`

機能説明

The `pppoed` daemon implements the server-side negotiation of PPPoE. When a client requests service from this daemon, a copy of `pppd(1M)` is invoked to handle the actual PPP communication.

At startup, options are read from the command line and the `/etc/ppp/pppoe` file. After these options have been read, options in the per-device `/etc/ppp/pppoe.`*device* files are read, using the device names specified on the command line or in `/etc/ppp/pppoe`. Device names are not permitted in the per-device files. It is not an error if any of these files are absent; missing files are ignored.

Options are reread in the same order on `SIGHUP`. Except for the possibility of short delays due to the processing time, `SIGHUP` does not interfere with any client operations. Current status, including options read, is dumped to `/tmp/pppoed.`*pid* on `SIGINT`.

The options are used to set up a list of services to be offered to PPPoE clients on the broadcast domains (Ethernet subnets) specified by the named devices. Option parsing is always in one of two modes, either global mode or service mode. The initial mode at the beginning of each file (and the command line) is global mode. Options specified in global mode serve as default values for subsequently defined services. Service mode is entered by the `service` *name* option. In this mode, the named option is defined. Options that appear in this mode override any global mode definitions for the current service.

The option parsing follows standard shell tokenizing rules, using whitespace to delimit tokens, quotes to enclose strings that can contain whitespace, and escape sequences for special characters. Environment variables are substituted using familiar `$VAR` and `${VAR}` syntax and set using `NEWVAR=`*string*. Variables are both usable in subsequent options and provided to the `pppd(1M)` processes spawned for each client, but they are interpreted as they are encountered during option processing. Thus, all set variables are seen by all processes spawned; position in the configuration files has no effect on this.

オプション

The `pppoed` daemon supports the following options:

`client [except]` *client-list*

This option restricts the clients that may receive the service. If the `except` keyword is given, then the clients on the list cannot access the service, but others can. If this keyword is not given, then only the listed clients can access the service.

This option can be specified more than once for a given service. For a given client, first match among all listed options encountered specifies the handling. If it matches an option with `except` specified, then access is denied. Otherwise, it is granted. The `client` list within a service is prepended to any list specified in the global context.

If no `client` options are given or if all options are specified with `except`, then all clients are permitted by default. If any `client` options without `except` are specified, then no clients are permitted by default.

The *client-list* is a comma-separated list of client identifiers. The match is made if any client on the list matches; thus, these are logically "ORed" together. Each client identifier can be either a symbolic name (resolved through `/etc/ethers` or NIS, as defined by `/etc/nsswitch.conf`) or a hexadecimal Ethernet address in the format `x:x:x:x:x:x`. In the latter case, any byte of the address can be "`*`", which matches any value in that position. For example, `40:0:1a:*:*:*` matches Ethernet adapters from the manufacturer assigned block `40:0:1a`.

`debug`

Increase debug logging detail level by one. The detail levels are 0 (no logging), 1 (errors only; the default), 2 (warnings), 3 (informational messages), and 4 (debug messages). Log messages are written by default to `syslog(3C)` using facility *daemon* (see the `log` option below). When specified on the command line or in the global context of the `/etc/ppp/pppoe` file, this option also sets the daemon's default (non-service-related) detail level.

`device` *device-list*

Specify the devices on which the service is available. The *device-list* is a comma-separated list of logical device names (without the leading `/dev/`), such as `hme0`. This option is ignored if encountered in the per-device `/etc/ppp/pppoe.`*device* files.

`extra` *string*

Specifies extra options to `pppd(1M)`. It defaults to "`plugin pppoe.so directtty`" and usually does not need to be overridden.

`file` *path*

Suspends parsing of the current file, returns to global mode, and reads options from *path*. This file must be present and readable; if it is not, an error is logged. When the end of that file is reached, processing returns to the current file and the mode is reset to global again.

The global mode options specified in files read by this command use the options set in the current file's global mode; this condition extends to any file included by those files. All files read are parsed as though the command line had specified this option, and thus inherit the command line's global modes.

This option can be used to revert to global mode at any point in an option file by specifying `file /dev/null`.

`group` *name*
: Specifies the group ID (symbolic or numeric) under which `pppd` is executed. If `pppoed` is not run as root, this option is ignored.

`log` *path*
: Specifies an alternate debug logging file. Debug messages are sent to this file instead of `syslog`. The special name `syslog` is recognized to switch logging back to `syslog`. When specified on the command line or in the global context of the `/etc/ppp/pppoe` file, this option also sets the daemon's default (non-service-related) log file.

`nodebug`
: Set debug logging detail level to 0 (no logging). When specified on the command line or in the global context of the `/etc/ppp/pppoe` file, this option also sets the daemon's default (non-service-related) detail level.

`nowildcard`
: Specifies that the current service should not be included in response to clients requesting "any" service. The client must ask for this service by name. When specified on the command line or in the global context of the `/etc/ppp/pppoe` file, this option causes `pppoed` to ignore all wildcard service requests.

`path` *path*
: Specifies the path to the `pppd` executable. Defaults to `/usr/bin/pppd`.

`pppd` *string*
: Passes command-line arguments to `pppd`. It can be used to set the IP addresses or configure security for the session. The default value is the empty string.

`server` *string*
: Specifies the PPPoE Access Concentrator name to be sent to the client. It defaults to "Solaris PPPoE".

`service` *name*
: Closes any service being defined and begins definition of a new service. The same service name can be used without conflict on multiple devices. If the same service name is used on a single device, then the last definition encountered during parsing overrides all previous definitions.

`user` *name*
: Specifies the user ID, symbolic or numeric, under which `pppd` is executed. If `pppoed` is not run as root, this option is ignored.

`wildcard`
: Specifies that the service should be included in responses to client queries that request "any" service, which is done by requesting a service name of length zero. When specified on the command line or in the global context of the

/etc/ppp/pppoe file, this option causes pppoed to ignore all wildcard service requests. This is the default.

使用例

例 1 Configuring for Particular Services

In the /etc/ppp/pppoe file:

```
service internet
    device $DEV
    pppd "proxyarp 192.168.1.1:"
service debugging
    device hme0,$DEV
    pppd "debug proxyarp 192.168.1.1:"
```

You then invoke the daemon with:

```
example% /usr/lib/inet/pppoed DEV=eri0
```

The lines in /etc/ppp/pppoe and the preceding command result in offering services "internet" and "debugging" (and responding to wildcard queries) on interface eri0, and offering only service "debugging" on interface hme0.

Signals

The pppoed daemon responds to the following signals:

SIGHUP Causes pppoed to reparse the original command line and all configuration files, and close and reopen any log files.

SIGINT Causes a snapshot of the state of the pppoed daemon to be written to /tmp/pppoed.*pid* (where *pid* is the decimal process ID of the daemon).

ファイル

/usr/lib/inet/pppoed	executable command
/dev/sppptun	Solaris PPP tunneling device driver
/etc/ppp/pppoe	main configuration option file
/etc/ppp/pppoe.*device*	per–device configuration option file
/etc/ppp/pppoe-errors	location of output from pppd's stderr
/etc/ppp/pppoe.if	list of Ethernet interfaces to be plumbed at boot time
/tmp/pppoed.*pid*	ASCII text file containing dumped pppoed state information

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ppp/tunnel

関連項目

pppd(1M), pppoec(1M), sppptun(1M), sppptun(7M)

Mamakos, L., et al. RFC 2516, A Method for Transmitting PPP Over Ethernet (PPPoE). Network Working Group. February 1999

注意事項

Because pppd is installed setuid root, this daemon need not be run as root. However, if it is not run as root, the user and group options are ignored.

The Ethernet interfaces to be used must be plumbed for PPPoE using the sppptun(1M) utility before services can be offered.

The daemon operate runs even if there are no services to offer. If you want to modify a configuration, it is not necessary to terminate the daemon. Simply use pkill -HUP pppoed after updating the configuration files.

The PPPoE protocol is far from perfect. Because it runs directly over Ethernet, there is no possibility of security and the MTU is limited to 1492 (violating RFC 1661's default value of 1500). It is also not possible to run the client and the server of a given session on a single machine with a single Ethernet interface for testing purposes. The client and server portions of a single session must be run on separate Ethernet interfaces with different MAC addresses.

名前

pppstats – print PPP statistics

形式

```
pppstats [-a] [-v] [-r] [-z] [-c <count>] [-w <secs>]
      [interface]
```

機能説明

The `pppstats` utility reports PPP-related statistics at regular intervals for the specified PPP interface. If the interface is unspecified, `pppstats` defaults to `sppp0`. The display is split horizontally into input and output sections containing columns of statistics describing the properties and volume of packets received and transmitted by the interface.

オプション

The `pppstats` options are:

- `-a` Display absolute values rather than deltas. With this option, all reports show statistics for the time elapsed since the link was initiated. Without this option, the second and subsequent reports show statistics for the time since the last report.
- `-c` *count* Repeat the display *count* times. If this option is not specified, the default repeat count is 1 if the `-w` option is not specified, otherwise infinity.
- `-r` Display additional statistics summarizing the compression ratio achieved by the packet compression algorithm in use.
- `-v` Display additional statistics relating to the performance of the Van Jacobson TCP header compression algorithm.
- `-w` *wait* Pause *wait* seconds between each display. If this option is not specified, the default interval is five seconds.
- `-z` Instead of the standard display, show statistics indicating the performance of the packet compression algorithm in use.

拡張機能説明

The following fields are printed on the input side when the `-z` option is not used:

- `IN` Total number of bytes received by this interface.
- `PACK` Total number of packets received by this interface.
- `VJCOMP` Number of header-compressed TCP packets received by this interface.
- `VJUNC` Number of header-uncompressed TCP packets received by this interface. Not reported when the `-r` option is specified.
- `VJERR` Number of corrupted or bogus header-compressed TCP packets received by this interface. Not reported when the `-r` option is specified.
- `VJTOSS` Number of VJ header-compressed TCP packets dropped on reception by this interface because of preceding errors. Only reported when the `-v` option is specified.
- `NON-VJ` Total number of non-TCP packets received by this interface. Only reported when the `-v` option is specified.

RATIO Compression ratio achieved for received packets by the packet compression scheme in use, defined as the uncompressed size divided by the compressed size. Only reported when the -r option is specified.

UBYTE Total number of bytes received, after decompression of compressed packets. Only reported when the -r option is specified.

The following fields are printed on the output side:

OUT Total number of bytes transmitted from this interface.

PACK Total number of packets transmitted from this interface.

VJCOMP Number of TCP packets transmitted from this interface with VJ-compressed TCP headers.

VJUNC Number of TCP packets transmitted from this interface with VJ-uncompressed TCP headers. Not reported when the -r option is specified.

NON-VJ Total number of non-TCP packets transmitted from this interface. Not reported when the -r option is specified.

VJSRCH Number of searches for the cached header entry for a VJ header compressed TCP packet. Only reported when the -v option is specified.

VJMISS Number of failed searches for the cached header entry for a VJ header compressed TCP packet. Only reported when the -v option is specified.

RATIO Compression ratio achieved for transmitted packets by the packet compression scheme in use, defined as the size before compression divided by the compressed size. Only reported when the -r option is specified.

UBYTE Total number of bytes to be transmitted before packet compression is applied. Only reported when the -r option is specified.

When the -z option is specified, pppstats displays the following fields relating to the packet compression algorithm currently in use. If packet compression is not in use, these fields display zeroes. The fields displayed on the input side are:

COMPRESSED BYTE Number of bytes of compressed packets received.

COMPRESSED PACK Number of compressed packets received.

INCOMPRESSIBLE BYTE Number of bytes of incompressible packets (that is, those which were transmitted in uncompressed form) received.

INCOMPRESSIBLE PACK Number of incompressible packets received.

COMP RATIO Recent compression ratio for incoming packets, defined as the uncompressed size divided by the compressed size (including both compressible and incompressible packets).

The fields displayed on the output side are:

COMPRESSED BYTE	Number of bytes of compressed packets transmitted.
COMPRESSED PACK	Number of compressed packets transmitted.
INCOMPRESSIBLE BYTE	Number of bytes of incompressible packets received; that is, those that were transmitted by the peer in uncompressed form.
INCOMPRESSIBLE PACK	Number of incompressible packets transmitted.
COMP RATIO	Recent compression ratio for outgoing packets.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ppp
Interface Stability	Committed

関連項目

pppd(1M), attributes(5)

名前

praudit – print contents of an audit trail file

形式

`praudit [-lrsx] [-ddel] [filename]...`

機能説明

`praudit` reads the listed *filenames* (or standard input, if no *filename* is specified) and interprets the data as audit trail records as defined in `audit.log(4)`. By default, times, user and group IDs (UIDs and GIDs, respectively) are converted to their ASCII representation. Record type and event fields are converted to their ASCII representation. A maximum of 100 audit files can be specified on the command line.

オプション

The following options are supported:

`-ddel`
: Use *del* as the field delimiter instead of the default delimiter, which is the comma. If *del* has special meaning for the shell, it must be quoted. The maximum size of a delimiter is three characters. The delimiter is not meaningful and is not used when the `-x` option is specified.

`-l`
: Print one line per record.

`-r`
: Print records in their raw form. Times, UIDs, GIDs, record types, and events are displayed as integers. This option is useful when naming services are offline. The `-r` option and the `-s` option are exclusive. If both are used, a format usage error message is output.

`-s`
: Display records in their short form. Numeric fields' ASCII equivalents are looked up by means of the sources specified in the `/etc/nsswitch.conf` file (see `nsswitch.conf(4)`). All numeric fields are converted to ASCII and then displayed. The short ASCII representations for the record type and event fields are used. This option and the `-r` option are exclusive. If both are used, a format usage error message is output.

`-x`
: Print records in XML form. Tags are included in the output to identify tokens and fields within tokens. Output begins with a valid XML prolog, which includes identification of the DTD which can be used to parse the XML.

ファイル

`/etc/security/audit_event`
: Audit event definition and class mappings.

`/etc/security/audit_class`
: Audit class definitions.

`/usr/share/lib/xml/dtd`
: Directory containing the verisioned DTD file referenced in XML output, for example, `adt_record.dtd.1`.

`/usr/share/lib/xml/style`
: Directory containing the versioned XSL file referenced in XML output, for example, `adt_record.xsl.1`.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

The command stability is evolving. The output format is unstable.

関連項目

getent(1M), getpwuid(3C), gethostbyaddr(3NSL), ethers(3SOCKET), getipnodebyaddr(3SOCKET), audit.log(4), audit_class(4), audit_event(4), group(4), nsswitch.conf(4), passwd(4), attributes(5)

See the section on Solaris Auditing in 『Oracle Solaris の管理: セキュリティーサービス』.

名前

projadd – administer a new project on the system

形式

```
projadd [-n] [-f filename] [-p projid [-o]] [-c comment]
      [-U user [,user]... ] [-G group [,group]... ]
      [ [-K name [=value [,value]...]...]] project
```

機能説明

`projadd` adds a new project entry to the `/etc/project` file. If the files backend is being used for the project database, the new project is available for use immediately upon the completion of the `projadd` command.

オプション

The following options are supported:

`-c` *comment*

Add a project comment. Comments are stored in the project's entry in the `/etc/project` file. Generally, comments contain a short description of the project and are used as the field for the project's full name.

Specify *comment* as a text string. *comment* cannot contain a colon (`:`) or NEWLINE.

`-f` *filename*

Specify the project file to modify. If no *filename* is specified, the system project file, `/etc/project`, is modified.

`-G` *group*[,group...]

Specify a group list for the project.

`-K` *name*[=*value*[,*value*...]

Specify an attribute list for the project. Multiple `-K` options can be specified to set values on multiple keys, such as:

```
-K key1=value1 -K "key2=(value2a),(value2b)"
```

Resource control attributes use parentheses to specify values for a key. Because many user shells interpret parentheses as special characters, it is best to enclose an argument to `-K` that contains parentheses with double quotes, as shown above and in EXAMPLES, below. See `resource_controls(5)` for a description of the resource controls you can specify for a project.

`-n`

Syntax check. Check the format of the existing system project file and modifications only. The contents of the existing project file, such as user names, group names, and resources that are specified in the project attributes are not checked.

`-o`

This option allows the project ID specified by the `-p` option to be non-unique within the project file.

`-p` *projid*

Set the project ID of the new project.

Specify *projid* as a non-negative decimal integer below `UID_MAX` as defined in `limits.h`. *projid* defaults to the next available

unique number above the highest number currently assigned. For example, if *projid*s `100`, `105`, and `200` are assigned, the next default *projid* is `201`. *projid*s between `0-99` are reserved by SunOS.

`-U` *user*[,user...] Specify a user list for the project.

オペランド

The following operands are supported:

project The name of the project to create. The *project* operand is a string consisting of characters from the set of alphabetic characters, numeric characters, underline (_), and hyphen (-). The period ('.') is reserved for projects with special meaning to the operating system. The first character of the project name must be a letter. An error message is displayed if these restrictions are not met.

使用例

例 1 Adding a Project

The following command creates the project `salesaudit` and sets the resource controls specified as arguments to the `-K` option.

```
projadd -p 111 -G sales,finance -c "Auditing Project" \
   -K "rcap.max-rss=10GB" \
   -K "process.max-file-size=(priv,50MB,deny)" \
   -K "task.max-lwps=(priv,100,deny)" salesaudit
```

This command would produce the following entry in `/etc/project`:

```
salesaudit:111:Auditing Project::sales,finance: \
process.max-file-size=(priv,52428800,deny); \
rcap.max-rss=10737418240;task.max-lwps=(priv,100,deny)
```

Note that the preceding would appear as one line in `/etc/project`.

Comparing the `projadd` command and resulting output in `/etc/project`, note the effect of the scaling factor in the resource cap (`rcap.max-rss=10GB`) and the resource control (`process.max-file-size=(priv,50MB,deny)`). Modifiers, such as B, KB, and MB, and scaling factors are specified in `resource_controls(5)`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`2` The command syntax was invalid. A usage message for `projadd` is displayed.

`3` An invalid argument was provided to an option.

`4` The *projid* given with the `-p` option is already in use.

`5` The project files contain an error. See `project(4)`.

`6` The project to be added, group, user, or resource does not exist.

9 The project is already in use.

10 Cannot update the /etc/project file.

ファイル

/etc/project System project file

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

Invocation is evolving. Human readable output is unstable.

関連項目

projects(1), groupadd(1M), groupdel(1M), groupmod(1M), grpck(1M), projdel(1M), projmod(1M), useradd(1M), userdel(1M), usermod(1M), project(4), attributes(5), resource_controls(5)

注意事項

In case of an error, projadd prints an error message and exits with a non-zero status.

projadd adds a project definition only on the local system. If a network name service such as NIS or LDAP is being used to supplement the local /etc/project file with additional entries, projadd cannot change information supplied by the network name service.

名前

projdel – delete a project from the system

形式

`projdel [-f filename] project`

機能説明

The `projdel` utility deletes a project from the system and makes the appropriate changes to the system file.

オプション

The following options are supported:

`-f` *filename* Specify the project file to modify. If no *filename* is specified, the system project file, `/etc/project`, is modified.

オペランド

The following operands are supported:

project The name of the project to be deleted.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`2` The command syntax was invalid. A usage message for `projdel` is displayed.

`3` An invalid argument was provided to an option.

`4` The *projid* given with the `-p` option is already in use.

`5` The project files contain an error. See `project(4)`.

`6` The project to be modified, group, user, or resource does not exist.

`9` The project is already in use.

`10` Cannot update the `/etc/project` file.

ファイル

`/etc/project` System project file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface stabilty	See below.

Invocation is evolving. Human readable output is unstable.

関連項目

`projects(1)`, `groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `grpck(1M)`, `logins(1M)`, `projadd(1M)`, `projmod(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `project(4)`, `attributes(5)`

診断

In case of an error, `projdel` prints an error message and exits with a non-zero status.

注意事項

`projdel` deletes a project definition only on the local system. If a network name service such as NIS or LDAP is being used to supplement the local `/etc/project` file with additional entries, `projdel` cannot change information supplied by the network name service.

名前

projmod – modify a project's information on the system

形式

```
projmod [-n] [-A|-f filename | -]

projmod [-n] [-A|-f filename | -] [-p projid [-o]]
     [-c comment] [-a|-s|-r] [-U user [,user]... ]
     [-G group [,group]... ]
     [ [-K name [=value [,value]...]...]]
     [-l new_projectname] project
```

機能説明

The `projmod` utility modifies a project's definition on the system. `projmod` changes the definition of the specified project and makes the appropriate project-related system file and file system changes.

オプション

The following options are supported:

`-A`
: Apply the project's resource controls, as defined in the system's project database, to the project if it is active.

`-a`
: Specify that the users, groups, attributes, or attribute values specified by the `-U`, `-G` or `-K` options should be added to the project, rather than replacing the existing member or attribute list.

`-c` *comment*
: Specify *comment* as a text string. Generally, *comment* contains a short description of the project. This information is stored in the project's `/etc/project` entry.

`-f` *filename* | –
: Specify the project file to modify or validate or specify input from stdin for validation. As noted under OPERANDS, if you do not specify a project in a `projmod` command line, `projmod` validates the argument to `-f`. If you do not use this option, the system project file, `/etc/project`, is modified.

`-G` *group* [,*group*...]
: Specify a replacement list of member groups of the project. When used in conjunction with the `-a` or `-r` options, this option specifies a list of groups to be added or removed from the project.

`-K` *name*[=*value*[,*value*...]
: Specify a replacement list of project attributes for the project. When used in conjunction with the `-a`, `-r`, or `-s` options, this option specifies a list of attribute values to be added, removed, or replaced in the project. Attributes must be delimited by semicolons (`;`). Multiple `-K` options can be specified to set, add, remove, or substitute values on multiple keys, such as:

  `-K` *key1=value1* `-K "`*key2*`=(`*value2a*`),(`*value2b*`)"`

  Resource control attributes use parentheses to specify values for a key. Because many user shells interpret parentheses as special

characters, it is best to enclose an argument to `-K` that contains parentheses with double quotes, as shown above and in EXAMPLES, below. See `resource_controls(5)` for a description of the resource controls you can specify for a project.

`-l` *new_projectname*
: Specify the new project name for the project. The *new_projectname* argument is a string consisting of characters from the set of alphabetic characters, numeric characters, period (`.`), underline (`_`), and hyphen (`-`). The first character should be alphabetic. An error message is written if these restrictions are not met. The project name must also be unique within the project file.

`-n`
: Syntax check. Check the format of the existing system project file and modifications only. The contents of the existing project file, such as user names, group names, and resources that are specified in the project attributes are not checked.

`-o`
: This option allows the project ID specified by the `-p` option to be non-unique within the project file.

`-p` *projid*
: Specify a new project ID for the project. It must be a non-negative decimal integer less than MAXUID as defined in `param.h`. This value must be unique within the project file if the `-o` option is not specified.

`-r`
: Specify that the users, groups, attributes, or attribute values specified by the `-U`, `-G` or `-K` options should be removed from the project, rather than replacing the existing member or attribute list.

`-s`
: Specify that the list of attributes specified by the `-K` option should have their values replaced. If the attributes do not exist, they are added as if the a option was used. This option has no effect the `-U` or `-G` options.

`-U` *user* [,*user*...]
: Specify a replacement list of member users of the project. When used in conjunction with the `-a` or `-r` options, this option specifies a list of users to be added or removed from the project.

オペランド

The following operands are supported:

project
: An existing project name to be modified or displayed.

(none)
: If no operand is given, the project file is validated without modifying any project.

使用例

例 1 Using the -K Option for Addition of an Attribute Value

Consider the following project(4) entry:

```
salesaudit:111:Auditing Project::sales,finance: \
  process.max-file-size=(priv,52428800,deny); \
  task.max-lwps=(priv,100,deny)
```

The preceding would appear as one line in /etc/project. For this and the following examples, the focus is on the attributes field in the project entry. That is, the last field, the field following the last semicolon.

The attributes field for the project salesaudit lists the following resource control:

```
task.max-lwps=(priv,1000,signal=KILL)
```

The following projmod command adds an action clause to the preceding entry:

```
# projmod -a -K "task.max-lwps=(priv,100,deny)" salesaudit
```

...with the resulting attributes field in the entry for salesaudit:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

例 2 Using the -K Option for the Substitution of an Attribute Value

Assume an attributes field in a project(4) entry for the project salesaudit that lists the following resource control:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

The following projmod command substitutes the action clause specified in the command for the action clauses in the preceding entry:

```
# projmod -s -K "task.max-lwps=(priv,500,signal=SIGSTOP)" salesaudit
```

...with the resulting attributes field in the entry for salesaudit:

```
task.max-lwps=(priv,500,signal=SIGSTOP)
```

例 3 Using the -K Option for Removal of an Attribute Value

Assume an attributes field in a project(4) entry for a project salesaudit that lists the following resource control:

```
task.max-lwps=(priv,100,deny),(priv,1000,signal=KILL)
```

The following projmod command removes the first action clause from the preceding entry:

```
# projmod -r -K "task.max-lwps=(priv,100,deny)" salesaudit
```

...with the resulting attributes field in the entry for salesaudit:

例 3 Using the -K Option for Removal of an Attribute Value *(続き)*

```
task.max-lwps=(priv,1000,signal=KILL)
```

例 4 Specifying Multiple Attribute Values

Suppose you want to achieve the following resource controls for the project `salesaudit`:

```
task.max-lwps=(priv,100,deny)
process.max-file-size=(priv,50MB,deny)
```

The following `projmod` command adds these resource controls for `salesaudit`:

```
# projmod -a -K "task.max-lwps=(priv,100,deny)" \
-K "process.max-file-size=(priv,50MB,deny)" salesaudit
```

...with the resulting attributes field in the entry for `salesaudit`:

```
task.max-lwps=(priv,100,deny);process.max-file-size=(priv,52428800,deny)
```

In this example, note the effect of the use of the modifier and scaling factor for the resource control `process.max-file-size`. The specification in `projmod`:

```
"process.max-file-size=(priv,50MB,deny)"
```

...becomes, in `/etc/project`:

```
process.max-file-size=(priv,52428800,deny)
```

That is, `50MB` is expanded to `52428800`. The modifiers, such as MB, and scaling factors you can use for resource controls are specified in `resource_controls(5)`.

例 5 Binding a Pool to a Project

The following command sets the `project.pool` attribute for the project `sales`.

```
# projmod -a -K project.pool=salespool sales
```

例 6 Evaluating Input from stdin

The following command uses the `-f` option without a project name operand to evaluate the contents of an NIS projects map.

```
# ypcat project | projmod -f -
```

終了ステータス

In case of an error, `projmod` prints an error message and exits with one of the following values:

The following exit values are returned:

`0` Successful completion.

`2` The command syntax was invalid. A usage message for `projmod` is displayed.

3 An invalid argument was provided to an option.

4 The *projid* given with the -p option is already in use.

5 The project files contain an error. See project(4).

6 The project to be modified, group, user, or resource does not exist.

9 The project is already in use.

10 Cannot update the /etc/project file.

ファイル

/etc/group System file containing group definitions

/etc/project System project file

/etc/passwd System password file

/etc/shadow System file containing users' encrypted passwords and related information

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

Invocation is evolving. Human readable output is unstable.

関連項目

groupadd(1M), groupdel(1M), groupmod(1M), projadd(1M), projdel(1M), useradd(1M), userdel(1M), usermod(1M), passwd(4), project(4), attributes(5), resource_controls(5)

注意事項

The projmod utility modifies project definitions only in the local /etc/project file. If a network name service such as NIS or LDAP is being used to supplement the local files with additional entries, projmod cannot change information supplied by the network name service. However projmod verifies the uniqueness of project name and project ID against the external name service.

名前

prstat – アクティブなプロセスの統計を報告する

形式

```
prstat [-acHJLmRrtTv] [-d u | d] [-C psrsetlist] [-h lgrplist]
     [-j projlist] [-k tasklist] [-n ntop[,nbottom]]
     [-p pidlist] [-P cpulist] [-s key | -S key ]
     [-u euidlist] [-U uidlist] [-z zoneidlist] [-Z]
     [interval [count]]
```

機能説明

prstat ユーティリティーは、システム上のアクティブなプロセスをすべて繰り返し検査し、選択した出力モードとソート順に基づいて統計を報告します。prstat には、指定した PID、UID、ゾーン ID、CPU ID、およびプロセッサセット ID と一致するプロセスだけを検査するオプションが用意されています。

-j、-k、-C、-p、-P、-u、-U、および -z オプションには、引数としてリストを指定できます。リストの項目は、コンマで区切られるか、あるいは引用符で囲まれてコンマかスペースで区切られます。

オプションを指定しない場合、prstat はすべてのプロセスを検査し、CPU 使用状況の順に統計を報告します。

オプション

次のオプションがサポートされています。

-a
: プロセスとユーザーに関する情報を報告します。このモードでは、prstat はプロセスとユーザーに関する別個のレポートを同時に表示します。

-c
: 前のレポートに重ねて出力するのではなく、前のレポートの下に新しいレポートを出力します。

-C *psrsetlist*
: 設定したリストのプロセッサセットにバインドされたプロセスまたは lwp のみを報告します。各プロセッサセットは、psrset(1M) によって報告される整数で識別されます。表示される平均負荷率は、指定したプロセッサセットの平均負荷率の合計です (pset_getloadavg(3C) を参照)。-L オプションが使用されていない場合でも、設定したリストのプロセッサセットにバインドされた 1 つ以上の LWP を持つプロセスが報告されます。

-d u | d
: 時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。 標準の日付フォーマットに d を指定します。date(1) を参照してください。

-h *lgrplist*
: 指定した *lgroup* のリストにホーム *lgroup* があるプロセスまたは lwp だけを報告します。無効な *lgroup* に対しては、プロセスまたは lwp は一覧表示されません。

-H
: ホーム *lgroup* に関する情報を報告します。このモードでは、prstat はプロセスまたは lwp のホーム *lgroup* を示す列を追加し、ヘッダーに LGRP と表示します。

-j *projlist*
指定したリストにプロジェクト ID があるプロセスまたは lwp だけを報告します。各プロジェクト ID はプロジェクト名または数値のプロジェクト ID のどちらでも指定できます。project(4) を参照してください。

-J
プロセスとプロジェクトに関する情報を報告します。このモードでは、prstat はプロセスとプロジェクトに関する別個のレポートを同時に表示します。

-k *tasklist*
tasklist にタスク ID があるプロセスまたは lwp だけを報告します。

-L
軽量プロセス (LWP) ごとの統計を報告します。デフォルトでは、prstat は各プロセスの LWP の数だけを報告します。

-m
マイクロステートプロセスアカウンティング情報を報告します。-v モードで表示されるすべてのフィールドに加え、このモードには、プロセスがシステムトラップ、テキストページフォルト、データページフォルトの処理に費やした時間、ユーザーロックの待機および CPU の待機 (待ち時間) に費やした時間の割合 (パーセント) も含まれています。

-n *ntop*[,*nbottom*]
出力の行数を制限します。*ntop* 引数は、プロセスまたは lwp 統計が報告される行数を決定し、*nbottom* 引数は、-a、-t、-T、-J、または -Z オプションが指定されている場合に、ユーザー、タスク、またはプロジェクト統計が報告される行数を決定します。デフォルトでは、prstat は、ウィンドウまたは端末に合わせた出力の行数を表示します。-c オプションを指定した場合、またはファイルへの出力を指定した場合、ntop および nbottom のデフォルト値は 15 および 5 です。

-p *pidlist*
指定したリストにプロセス ID があるプロセスだけを報告します。

-P *cpulist*
指定したリストにある CPU で最後に実行されたプロセスまたは lwp だけを報告します。各 CPU は、psrinfo(1M) によって報告される整数で識別されます。

-R
prstat をリアルタイムスケジューリングクラスに配置します。このオプションが使用される場合、prstat は、タイムシェアリングプロセスおよび対話型プロセスよりも優先されます。このオプションは、スーパーユーザーだけが使用できます。

-r
ユーザー名およびプロジェクト名の検索を無効にします。(これは -j、-u、または -U オプションでの検索には適用されません)。

-s *key*
出力行 (つまり、プロセス、lwp、またはユーザー) を *key* で降順にソートします。引数として使用できるのは 1 つの *key* だけです。

使用可能なキー値は 5 つあります。

cpu
プロセスの CPU の使用状況順にソートします。これはデフォルトです。

pri
プロセスの優先度順にソートします。

rss
常駐の設定サイズ順にソートします。

size
プロセスイメージのサイズ順にソートします。

time
プロセスの実行時間順にソートします。

-S *key*
出力行を *key* で昇順にソートします。使用可能な *key* 値は、-s オプションの場合と同じです。-s を参照してください。

-t
ユーザーごとの全体的な使用状況の要約を報告します。要約には、ユーザーが所有するプロセスまたは LWP の合計数、プロセスイメージの合計サイズ、常駐の合計設定サイズ、CPU の合計時間、および最近の CPU 時間とシステムメモリーの割合 (パーセント) が含まれます。

-T
プロセスとタスクに関する情報を報告します。このモードでは、prstat はプロセスとタスクに関する別個のレポートを同時に表示します。

-u *euidlist*
指定したリストに実効ユーザー ID があるプロセスだけを報告します。各ユーザー ID はログイン名または数値のユーザー ID のどちらでも指定できます。

-U *uidlist*
指定したリストに実ユーザー ID があるプロセスだけを報告します。各ユーザー ID はログイン名または数値のユーザー ID のどちらでも指定できます。

-v
プロセスの詳細な使用状況を報告します。この出力形式には、プロセスがユーザーモード、システムモード、および休止状態で費やした時間の割合 (パーセント) が含まれます。また、コンテキストの自発的および強制的な切り替えの数、システムコール、および受信したシグナルの数も含まれます。報告されない統計は、- 記号でマークされます。

-z *zoneidlist*
指定したリストにゾーン ID があるプロセスまたは LWP だけを報告します。各ゾーン ID は、ゾーン名または数値のゾーン ID のどちらでも指定できます。zones(5) を参照してください。

-Z
プロセスとゾーンに関する情報を報告します。このモードでは、prstat はプロセスとゾーンに関する別個のレポートを同時に表示します。

出力

次に、prstat レポートの列ヘッダーとその意味について説明します。

PID
プロセスのプロセス ID。

USERNAME
実ユーザー (ログイン) 名または実ユーザー ID。

SWAP
各ユーザー、プロジェクト、タスク、またはゾーンに関連付けられているプロセスのスワップ予約の合計容量。これは、ユーザー、プロジェクト、タスク、またはゾーンごとに、共有メモリーを一度だけ加算します。スワップが予約されるのは、匿名メモリーが割り当てられた場合、またはファイルが非公開にマップされた場合です。スワップの値は、キロバイト (K)、メガバイト (M)、またはギガバイト (G) で表されます。

RSS
プロセスの常駐の設定サイズ (RSS) (単位: K バイト (K)、M バイト (M)、または G バイト (G))。RSS 値は、proc(4) によって提供される見積りですが、これは実際の常駐設定サイズより少なく見積もることがあります。容量を計画するためにより正確な使用状況の情報を取得するには、代わりに pmap(1) で -x オプションを使用してください。

STATE
プロセスの状態

cpu*N*
プロセスは、CPU *N* 上で実行されています。

sleep
休止状態。 プロセスは、イベントが完了するのを待っている

wait
待機状態。プロセスは、CPU 使用率が CPU キャップの強制された制限値まで下がるのを待っている。resource_controls(5) の CPU-caps の説明を参照してください。

run
実行可能状態。プロセスは、実行待ち行列上にあります。

zombie
ゾンビ状態。 プロセスは終了していて、親プロセスは待っていない

stop
プロセスは停止されています。

PRI
プロセスの優先順位。数値が大きいほど優先順位が高くなります。

NICE
優先順位の計算に使用される nice 値。特定のスケージューリングクラスのプロセスのみが nice 値を持ちます。

TIME
プロセスの累積実行時間。

CPU
プロセスによって使用された最近の CPU 時間の割合 (パーセント)。非大域ゾーンで実行中で、プール機能がアクティブな場合、割合は、ゾーンがバインドされたプールによって使用中のプロセッサセット内にあるプロセッサの割合になります。

PROCESS
プロセスの名前 (実行されたファイルの名前)。

LWPID
報告される `lwp` の `lwp` ID。

NLWP
プロセス内の `lwp` の数。

一部のオプションを使用すると、前述のいくつかの列ヘッダーに加えて、次の列ヘッダーが表示されます。

NPROC
指定したコレクションにあるプロセスの数。

MEMORY
指定したプロセスコレクションで使用されるメモリーの割合 (パーセント)。

`-v` または `-m` オプションを指定すると、次の列が表示されます。

USR
プロセスがユーザーモードで費やした時間の割合 (パーセント)。

SYS
プロセスがシステムモードで費やした時間の割合 (パーセント)。

TRP
プロセスがシステムトラップの処理に費やした時間の割合 (パーセント)。

TFL
プロセスがテキストページフォルトの処理に費やした時間の割合 (パーセント)。

DFL
プロセスがデータページフォルトの処理に費やした時間の割合 (パーセント)。

LCK
プロセスがユーザーロックの待機に費やした時間の割合 (パーセント)。

SLP
プロセスが休止状態で費やした時間の割合 (パーセント)。

LAT
プロセスが CPU の待機に費やした時間の割合 (パーセント)。

VCX
コンテキストの自主的な切り替え数。

ICX
コンテキストの強制的な切り替え数。

SCL
システムコールの数。

SIG
受信されたシグナルの数。

`-L` オプションを指定すると、プロセス中の `lwp` ごとに 1 行ずつ出力され、一部のレポートフィールドにはプロセスではなく `lwp` の値が表示されます。

`-H` オプションを指定すると、次の列が表示されます。

LGRP
プロセスまたは lwp のホーム *lgroup*。

オペランド

次のオペランドがサポートされています。

count
統計を繰り返す回数を指定します。デフォルトでは、`prstat` は終了シグナルが受信されるまで統計を報告します。

interval
サンプリング間隔を秒単位で指定します。デフォルトの間隔は 5 秒です。

使用例

例 1 もっともアクティブな 5 つのスーパーユーザープロセスの報告

次のコマンドは、`CPU1` および `CPU2` で実行中のもっともアクティブな 5 つのスーパーユーザープロセスを報告します。

```
example% prstat -u root -n 5 -P 1,2 1 1

PID   USERNAME  SWAP   RSS STATE  PRI  NICE      TIME  CPU PROCESS/LWP
```

例 1　もっともアクティブな 5 つのスーパーユーザープロセスの報告　　*(続き)*

```
 306   root     3024K 1448K sleep   58    0   0:00.00 0.3% sendmail/1
 102   root     1600K  592K sleep   59    0   0:00.00 0.1% in.rdisc/1
 250   root     1000K  552K sleep   58    0   0:00.00 0.0% utmpd/1
 288   root     1720K 1032K sleep   58    0   0:00.00 0.0% sac/1
   1   root      744K  168K sleep   58    0   0:00.00 0.0% init/1
TOTAL:       25, load averages:  0.05, 0.08, 0.12
```

例 2　プロセス使用状況の詳細情報の表示

次のコマンドは、ユーザー root および john が所有する最小の常駐設定サイズのプロセスに関するプロセス使用状況の詳細情報を表示します。

```
example% prstat -S rss -n 5 -vc -u root,john

 PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWP
   1 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 init/1
 102 root     0.0 0.0  -   -   -   -  100  -    0   0   3   0 in.rdisc/1
 250 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 utmpd/1
1185 john    0.0 0.0  -   -   -   -  100  -    0   0   0   0 csh/1
 240 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 switchd/4
 TOTAL:       71, load averages:  0.02, 0.04, 0.08
```

終了ステータス

次の終了値が返されます。

0
　正常終了。

1
　エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

date(1), lgrpinfo(1), plgrp(1), proc(1), ps(1), time(2), psrinfo(1M), psrset(1M), sar(1M), pset_getloadavg(3C), proc(4), project(4), attributes(5), resource_controls(5), zones(5)

注意事項

prstat によって表示されるシステム使用状況のスナップショットは、ある瞬間だけに当てはまるものであり、表示された時点では正確でない場合があります。-m オプションが指定された場合、prstat はプロセスごとにマイクロステートアカウンティングをオンに設定しようとします。prstat が終了すると、元の状態が復元されます。マイクロステートアカウンティング機能については、proc(4) を参照してください。

プロセスのグループの SWAP および RSS 列に報告される合計メモリーサイズは、共有メモリーセグメントを持つプロセスが使用する実際のメモリーサイズより多く見積もられる場合があります。

名前　prtconf – システム構成の出力

形式

```
/usr/sbin/prtconf [-V] | [-F] | [-x] | [-bpv] | [-acdDlPuv]
      [dev_path]
```

機能説明

prtconf コマンドはシステム構成情報を出力します。この出力には、メモリーの総量やシステム周辺機器の構成 (デバイスツリー形式) が含まれます。

コマンド行にデバイスパスを指定した場合、prtconf は、そのデバイスノードに関する情報だけを表示します。

オプション

次のオプションがサポートされています。

-a　コマンド行に指定したデバイスの祖先であるデバイスノードをすべて (つまり、ルートノードまで) 表示します。

-b　プラットフォームを特定するため、ファームウェアのデバイスツリーのルート設定を表示します。これらの設定の内容は「name」、「compatible」、「banner-name」、および「model」です。

-c　コマンド行に指定したデバイスノードをルートとするデバイスサブツリーを表示します。つまり、コマンド行に指定したデバイスノードの子孫であるデバイスノードをすべて表示します。

-d　ノード名に加え、PCI および PCI Express デバイスのベンダー ID とデバイス ID を表示します。

-D　デバイスツリー内のシステム周辺機器ごとに、周辺機器を管理するために使用されているデバイスドライバの名前を表示します。

-l　デバイスノードに関連付けられた /dev/chassis の場所を表示します。-v が使用されると、-l も暗黙的に指定されます。

-F　コンソールフレームバッファが存在する場合、そのデバイスパス名を返します。フレームバッファが存在しない場合、prtconf はゼロ以外の終了ステータスを返します。このフラグは単独で使用する必要があります。prtconf は、コンソールの名前、フレームバッファのデバイス、またはゼロ以外の終了ステータスだけを返します。たとえば、SUNW, Ultra-30 上のコンソールフレームバッファが ffb であれば、prtconf は次の値を返します。/SUNW,ffb@1e,0:ffb0。このオプションは、/dev/fb から実際のコンソールデバイスへのシンボリックリンクを作成するときに使用できます。

-p　SPARC プラットフォーム上のファームウェア (PROM) または x86 プラットフォーム上のブートシステムが提供するデバイスツリーから取得した情報を表示します。このオプションを使用して表示されるデバイスツリー情報は初期構成のスナップショットであり、後で行われた構成変更は正確に反映されない場合があります。

-P　疑似デバイスの情報も出力します。デフォルトでは、疑似デバイスの情報は出力されません。

`-u` `-v` とともに使用して、デバイスごとの情報を表示します。ベンダーおよび管理リストのプロパティーがある場合はそれを一覧表示します。

`-v` 詳細表示モードを指定します。

`-V` プラットフォームに固有な PROM (SPARC プラットフォーム) またはブートシステム (x86 プラットフォーム) のバージョン情報を表示します。このフラグは単独で使用する必要があります。出力は文字列です。文字列の形式は決まっておらず、プラットフォームに固有です。

`-x` 当該システム上のファームウェアが 64 ビット対応であるかどうかを報告します。既存のプラットフォームの中には、64 ビットカーネルを実行するためにファームウェアをアップグレードする必要があるものもあります。この操作が当該プラットフォームで適用されない場合、つまり、ファームウェアがすでに 64 ビット対応である場合、prtconf は何も表示せずに、ゼロの終了ステータスを返して終了します。この操作が当該プラットフォームで適用される場合、つまり、ファームウェアが 64 ビット対応でない場合、prtconf は標準出力に説明メッセージを表示して、ゼロ以外の終了ステータスを返して終了します。64 ビットカーネルを実行するためにファームウェアをアップグレードする必要があるかどうかの詳細については、プラットフォームのハードウェアドキュメントを参照してください。

このフラグは他のすべてのフラグを無効にするので、単独で使用する必要があります。

オペランド

次のオペランドがサポートされています。

dev_path ターゲットとなるデバイスマイナーノード、デバイスへのパス、または、デバイスリンクへのパスです。これらデバイスノードの構成情報が表示されます。

終了ステータス

次の終了値が返されます。

`0` 正常終了

`0` 以外 `-F` オプションを指定した場合 (SPARC のみ)、ゼロ以外の戻り値は、出力デバイスがフレームバッファではないことを意味します。`-x` オプションを指定した場合、ゼロ以外の戻り値は、ファームウェアが 64 ビット対応ではないことを意味します。それ以外の場合には、ゼロ以外の戻り値はエラーが発生したことを意味します。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os

属性タイプ	属性値
インタフェースの安定性	不確実

関連項目

fuser(1M), modinfo(1M), sysdef(1M), driver(4), driver.conf(4), attributes(5)

『Sun ハードウェアマニュアル』

SPARC のみ

openprom(7D)

注意事項

prtconf コマンドの出力はシステムにインストールされている PROM のバージョンに大きく依存します。したがって、出力は潜在的にあらゆる状況の影響を受けます。

driver not attached というメッセージは、現在、デバイスの当該インスタンスにドライバが接続されていないことを意味します。一般に、ドライバは必要に応じてロードおよびインストールされ、(そして、ハードウェアのインスタンスに接続され)、デバイスが使用されないときはアンインストールとアンロードされます。

x86 プラットフォーム上で prtconf -vp を使用すると、prtconf -v の部分的な情報が表示されます。prtconf -vp の整数設定の値を正しく解釈するには、バイトスワップが必要になる場合があります。

名前

prtdiag – システムの診断情報の出力

形式

```
/usr/sbin/prtdiag [-v] [-l]
```

機能説明

prtdiag は、sun4u、sun4v、および x86 システム上で、システム構成および診断情報を表示します。

診断情報は、システム内で障害の発生した現場交換可能ユニット (FRU) を表示します。

prtdiag に必要なインタフェース、出力、およびディレクトリ階層における位置は、まだ確定されていません。将来のリリースで変更される可能性があります。

prtdiag は、Sun Enterprise 10000 サーバーで実行された場合は、診断情報と環境ステータスを表示しません。それらの情報を得るには、システムサービスプロセッサ(SSP)上の/var/opt/SUNWssp/adm/${SUNW_HOSTNAME}/messagesファイルを参照してください。

オプション

次のオプションがサポートされています。

-l　ログの出力。システムで障害またはエラーが発生した場合は、syslogd(1M) のみに対してこの情報を出力します。

-v　詳細モード。最近発生した AC 電源障害の時刻、最近発生した重大なハードウェアエラーの情報、および (必要に応じて) 環境ステータスを表示します。重大なハードウェアエラーの情報は、FRU を修理する場合や、詳細な診断を行う場合に有用です。

終了ステータス

次の終了値が返されます。

0　システム内で何も異常が検出されなかった。

1　システム内で異常が検出されました。

2　メモリー不足などの、内的な prtdiag のエラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/library/platform
インタフェースの安定性	不確実*

* 出力は不安定です。

関連項目

modinfo(1M), prtconf(1M), psrinfo(1M), sysdef(1M), syslogd(1M), attributes(5), openprom(7D)

注意事項

すべての Solaris プラットフォーム上で、すべての診断およびシステム情報が得られるわけではないため、これらの情報が `prtdiag` で表示されないことがあります。こうしたプラットフォームでは、システムコントローラを使用してより詳しい情報を得ることができます。

名前

prtdscp – DSCP IP アドレスの表示

形式

```
prtdscp [-v ]
prtdscp [-v ] -h
prtdscp [-v ] -d
prtdscp [-v ] -s
```

機能説明

prtdscp は、Domain to Service Processor Communications Protocol (DSCP) リンクに関連付けられた IP アドレスを表示します。引数を指定しない場合、prtdscp は、DSCP リンクの両端の IP アドレスを表示します。-s または -d オプションを使用すると、それぞれサービスプロセッサまたはドメインサイドのどちらかの IP アドレスを個別に表示できます。

オプション

次のオプションがサポートされています。

- -v　冗長モード。プログラム内の進捗状況に関する追加の詳細情報を標準エラー出力に出力します。
- -h　ヘルプ。プログラムの使用法の簡単な概要を出力して、終了します。その他のコマンド行引数はすべて無視されます。
- -d　ローカルドメインの IP アドレスのみを表示します。
- -s　リモートのサービスプロセッサの IP アドレスのみを表示します。

使用例

例 1　両方のアドレスを表示する

次の例では、ローカルドメインの IP アドレスとリモート SP の IP アドレスの両方を表示します。

```
# prtdscp
Domain Address: 192.168.103.2
SP Address: 192.168.103.1
```

例 2　ローカル IP アドレスを表示する

次の例は、ローカルドメインの IP アドレスを表示します。

```
# prtdscp -d
192.168.103.2
```

例 3　リモート IP アドレスを表示する

次の例は、リモート SP の IP アドレスを表示します。

```
# prtdscp -s
192.168.103.1
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/domain-service-processor-protocol/sparc-enterprise, SUNWdscpu.u
インタフェースの安定性	確実

関連項目

attributes(5)

名前

prtfru – システムまたはドメイン上の FRU に関する FRUID 固有の情報の出力

形式

`/usr/sbin/prtfru [-d] | [-clx] [container]`

機能説明

`prtfru` ユーティリティーは、システムまたはドメインから FRUID データを取得する場合に使用します。その出力は、各コンテナに対する FRU (現場交換可能ユニット) ツリー内のパスを反映したツリー構造です。コンテナが見つかった場合、そのコンテナからのデータもツリー状の構造で出力されます。

引数を付けずに `prtfru` を使用すると、FRU 階層とすべての FRUID コンテナデータが出力されます。`prtfru` は、ファイルにリダイレクトできる標準出力に出力されます。

オプション

次のオプションがサポートされています。

- `-c` コンテナとそのデータのみを出力します。このオプションでは、FRU ツリー階層は出力されません。
- `-d` 現在のレジストリの DTD を標準出力に出力します。
- `-l` FRU ツリー階層のみを出力します。このオプションでは、コンテナデータは出力されません。
- `-x` `prtfrureg.dtd` のシステム識別子 (SYSTEM) を使用した XML 形式で出力します。

オプション `-c` および `-l` を一緒に使用すると、コンテナのすべてのリストを取得できます。

オペランド

次のオペランドがサポートされています。

- *container* FRU 階層での特定のコンテナの名前。つまり、`-l` オプションで表示されるコンテナの名前またはパス/名前のどちらかです。

終了ステータス

次の終了値が返されます。

- `0` すべての情報が見つかり、正常に出力されました。
- `>0` エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/fru-id

関連項目

`fruadm(1M)`, `attributes(5)`

名前

prtpicl – PICL ツリーの出力

形式

`/usr/sbin/prtpicl [-c picl_class] [-v]`

機能説明

`prtpicl` コマンドは、PICL デーモンによって保守される PICL ツリーを出力します。`prtpicl` の出力には、ノードの名前と PICL クラスが含まれます。

オプション

次のオプションがサポートされています。

`-c` *picl_class*　指定された名前の PICL クラスのノードのみを出力します。

`-v`　詳細モードで出力します。詳細モードでは、`prtpicl` は各ノードのプロパティーと値のリストを出力します。詳細モードはデフォルトで無効になっています。

終了ステータス

次の終了値が返されます。

`0`　正常終了。

`0` 以外　エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/picl

関連項目

`picld(1M)`, `attributes(5)`

名前

prtvtoc – ディスクジオメトリおよびパーティション分割に関する情報の報告

形式

`prtvtoc [-fhs] [-t vfstab] [-m mnttab] device`

機能説明

`prtvtoc` コマンドを使用すれば、ラベルの内容を表示できます。このコマンドは、スーパーユーザーのみが使用できます。

device 名は、`/dev/rdsk/c?t?d?s2` の形式で raw デバイスのファイル名にすることも、`/dev/dsk/c?t?d?s2` の形式でブロック型デバイスのファイル名にすることもできます。

オプション

次のオプションがサポートされています。

`-f`	空き領域の開始ブロックアドレス、ブロック数、未使用パーティションなど、ディスク空き領域の情報を報告します。
`-h`	通常の出力からヘッダーを省略します。
`-m` *mnttab*	マウントされているファイルシステムのリストとして、`/etc/mnttab` の代わりに *mnttab* を使用します。
`-s`	通常の出力から列ヘッダーを除くすべてのヘッダーを省略します。
`-t` *vfstab*	ファイルシステムデフォルトのリストとして、`/etc/vfstab` の代わりに *vfstab* を使用します。

使用例

例 1　prtvtoc コマンドを使用する

次の例では、424 M バイトのハードディスクで `prtvtoc` コマンドを使用します。

```
example# prtvtoc /dev/rdsk/c0t3d0s2
* /dev/rdsk/c0t3d0s2 partition map
*
* Dimension:
*     512 bytes/sector
*      80 sectors/track
*       9 tracks/cylinder
*     720 sectors/cylinder
*    2500 cylinders
*    1151 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
* *                          First    Sector   Last
* Partition   Tag   Flags   Sector   Count    Sector   Mount Directory
     0         2     00          0    76320    76319   /
     1         3     01      76320   132480   208799
     2         5     00          0   828720   828719
     5         6     00     208800   131760   340559   /opt
```

例 1　prtvtoc コマンドを使用する　　*(続き)*

```
     6         4     00     340560   447120   787679   /usr
     7         8     00     787680    41040   828719   /export/home
example#
```

上記の Tag 列内のデータは、次のようなパーティションのタイプを示します。

名前	数値
UNASSIGNED	0x00
BOOT	0x01
ROOT	0x02
SWAP	0x03
USR	0x04
BACKUP	0x05
STAND	0x06
VAR	0x07
HOME	0x08
ALTSCTR	0x09
CACHE	0x0a
RESERVED	0x0b
SYSTEM	0x0c
BOOT	0x18

上記の Flags 列内のデータは、次のように、パーティションをマウントする方法を示します。

名前	数値
MOUNTABLE, READ AND WRITE	0x00
NOT MOUNTABLE	0x01
MOUNTABLE, READ ONLY	0x10

例 2 -f オプションを付けて prtvtoc コマンドを使用する

次の例では、424M バイトのハードディスクで、-f オプションを付けて prtvtoc コマンドを使用します。

```
example# prtvtoc -f /dev/rdsk/c0t3d0s2
FREE_START=0 FREE_SIZE=0 FREE_COUNT=0 FREE_PART=34
```

例 3 1T バイトを超えるディスクで prtvtoc コマンドを使用する

次の例では、1T バイトを超えるディスクで prtvtoc コマンドを使用します。

```
example# prtvtoc /dev/rdsk/c1t1d0s2
* /dev/rdsk/c1t1d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
* 3187630080 sectors
* 3187630013 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
0      2    00         34    262144    262177
1      3    01     262178    262144    524321
6      4    00     524322 3187089340 3187613661
8     11    00  3187613662     16384 318763004
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

devinfo(1M), fmthard(1M), format(1M), mount(1M), attributes(5)

警告

mount コマンドでは、「マウント不可」のビットは検査されません。

名前

psradm – プロセッサの動作ステータスの変更

形式

```
psradm -f | -i | -n | -s [-v] [-F] processor_id

psradm -a -f | -i | -n | -s [-v] [-F]
```

機能説明

psradm ユーティリティーはプロセッサの動作ステータスを変更します。プロセッサの有効な状態は、on-line、off-line、spare、faulted、および no-intr です。

on-line プロセッサは、LWP (軽量プロセス) を処理し、システム内の I/O デバイスから割り込むことができます。

off-line プロセッサは、どの LWP も処理しません。通常、システム内の I/O デバイスから off-line プロセッサに割り込むことはできません。一部のプロセッサ、または特定の状況では、off-line プロセッサに対する割り込みを無効にできない場合があります。したがって、off-line での実際の影響は、マシンごとに異なる場合があります。

スペアプロセッサは LWP を処理しません。スペアプロセッサは、システムの特権ユーザーによって、またはシステム状態の変更に応じてカーネルによって、on-line、off-line、または no-intr に移行できます。

障害の発生したプロセッサは、カーネルが一定期間プロセッサの動作を監視して識別します。特権ユーザーは、障害の発生したプロセッサの状態を、on-line、off-line、spare、または no-intr に設定できますが、その場合、強制オプションを使用する必要があります。

no-intr プロセッサは、LWP を処理しますが、I/O デバイスから割り込まれません。

プロセッサに結合された LWP がある場合、追加の -F オプションを使用しないかぎり、プロセッサを off-line または spare にすることはできません。-F オプションは、プロセッサの動作ステータスを変更する前に、このような LWP のプロセッサとの結合を削除します。一部のアーキテクチャーでは、特定のプロセッサを off-line または spare にできない場合があります。たとえば、そのプロセッサが提供するリソースにシステムが依存している場合です。

システム内の少なくとも 1 つのプロセッサが LWP を処理できる必要があります。少なくとも 1 つのプロセッサが割り込み可能であることも必要です。off-line または spare プロセッサは割り込み可能なので、1 つのプロセッサが no-intr でほかのプロセッサがすべて off-line または spare であるが 1 つまたは複数の割り込みを受け入れる運用システムを持つことは可能です。

指定したプロセッサのいずれかの電源が切られていた場合、psradm が 1 つまたは複数のプロセッサの電源を入れる場合があります。

PRIV_SYS_RES_CONFIG 特権を持つユーザーだけが、psradm ユーティリティーを使用できます。

オプション

次のオプションがサポートされています。

- `-a` すべてのプロセッサ、またはできるだけ多くのプロセッサで処理を実行します。
- `-f` 指定されたプロセッサを `off-line` にします。
- `-F` 指定された別の状態に強制的に切り換えます。指定されたプロセッサのうち 1 つまたは複数が障害状態になった場合に必要です。ほかの移行オプションが指定されなかった場合は、指定されたプロセッサを faulted に設定します。移行の強制は、`faulted`、`spare`、または `off-line` の状態に対してのみ行えます。プロセッサの状態の移行を強制することによってどのスレッドに影響が及ぶかを調べるために、管理者は `pbind(1M)` で `-Q` オプションを使用するようにお勧めします。
- `-i` 指定されたプロセッサを `no-intr` に設定します。
- `-n` 指定されたプロセッサを `on-line` にします。
- `-s` 指定されたプロセッサを spare にします。
- `-v` 操作の各試行の結果を示すメッセージを出力します。

オペランド

次のオペランドがサポートされています。

processor_id `on-line`、`off-line`、`spare`、または `no-intr` に設定するプロセッサのプロセッサ ID。

個々のプロセッサ番号 (たとえば `3`)、スペースで区切った複数のプロセッサ番号 (たとえば `1 2 3`)、プロセッサ番号の範囲 (たとえば `1-4`) として *processor_id* を指定します。また、範囲と (個々または複数の) *processor_id* を組み合わせることも可能です (たとえば、`1-3 5 7-8 9`)。

使用例

例 1 プロセッサを off-line に設定する

次の例では、プロセッサ 2 および 3 を `off-line` に設定します。

```
% psradm -f 2 3
```

例 2 プロセッサを no-intr に設定する

次の例では、プロセッサ 1 および 2 を `no-intr` に設定します。

```
% psradm -i 1 2
```

例 3 プロセッサを spare に設定する

次の例では、プロセッサ 1 と 2 のどちらかが faulted の状態になっている場合でも、両方のプロセッサを spare に設定します。

```
% psradm -F -s 1 2
```

例 4　すべてのプロセッサを on-line に設定する

```
% psradm -a -n
```

例 5　プロセッサを off-line に強制する

次の例では、プロセッサ 1 および 2 をオフラインに設定し、それらに結合されたプロセスからプロセッサ結合を削除します。

```
% psradm -F -f 1 2
```

終了ステータス

次の終了値が返されます。

`0`　正常終了。

`>0`　エラーが発生しました。

ファイル

`/etc/wtmpx`　プロセッサステータスの変更を記録したレコード

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

pbind(1M), psrinfo(1M), psrset(1M), p_online(2), processor_bind(2), attributes(5)

診断

`psradm: processor 4: Invalid argument`
指定されたプロセッサが構成内に存在しません。

`psradm: processor 3: Device busy`
指定されたプロセッサは、LWP が結合されているか、システムで最後の `on-line` プロセッサであるか、または一部の重要なサービスを提供するためにシステムで必要とされているので、`off-line` にすることができませんでした。

`psradm: processor 3: Device busy`
指定されたプロセッサは、システムで最後の割り込み可能なプロセッサであるか、またはシステムが必要とする割り込みサービスを提供できるシステムで唯一のプロセッサであるために、`no-intr` に設定できませんでした。

`psradm: processor 3: Device busy`
指定されたプロセッサの電源が切られており、プラットフォーム固有のリソースの一部が利用できないために電源を入れることができません。

`psradm: processor 0: Not owner`
ユーザーにはプロセッサステータスを変更する権限がありません。

`psradm: processor 2: Operation not supported`
指定されたプロセッサの電源が切られており、プラットフォームは個々のプロセッサの電源投入をサポートしていません。

名前

psrinfo – プロセッサに関する情報の表示

形式

```
psrinfo [-p] [-v] [processor_id]...
psrinfo [-p] -s processor_id
```

機能説明

psrinfo はプロセッサに関する情報を表示します。それぞれの物理プロセッサは、複数の仮想プロセッサをサポートできます。それぞれの仮想プロセッサは、固有の割り込み ID を持つエンティティーであり、独立したスレッドを実行できます。

processor_id オペランドを使用しない場合、psrinfo は、構成されたプロセッサごとに 1 行を使って、そのステータスがオンライン、割り込み不可 (no-intr で指定)、スペア、オフライン、障害、または電源切断のどれであるか、および最後にそのステータスが変更された時刻を表示します。processor_id オペランドを使用すると、特定のプロセッサに関する情報が表示されます。「オペランド」の項を参照してください。

オプション

次のオプションがサポートされています。

-s *processor_id*
: サイレントモード。指定したプロセッサが完全にオンラインの場合に 1 を表示します。指定したプロセッサが割り込み不可、スペア、オフライン、障害、または電源切断の状態である場合に 0 を表示します。

  シェルスクリプトで psrinfo を使用するときに、サイレントモードを使用します。

-p
: システム内の物理プロセッサの数を表示します。

  -v オプションと組み合わせた場合、それぞれの物理プロセッサに関する追加情報を報告します。

-v
: 冗長モード。プロセッサタイプ、浮動小数点ユニットタイプ、およびクロック速度など、指定したプロセッサに関する追加情報を表示します。この情報のいずれかを判定できない場合、psrinfo は unknown を表示します。

  -p オプションと組み合わせた場合、それぞれの物理プロセッサに関する追加情報を報告します。

オペランド

次のオペランドがサポートされています。

processor_id
: 情報を表示するプロセッサのプロセッサ ID。

  個々のプロセッサ番号 (たとえば 3)、スペースで区切った複数のプロセッサ番号 (たとえば 1 2 3)、プロセッサ番号の範囲 (たとえば 1-4) として *processor_id* を指定します。また、範囲と (個々または複数の) *processor_id* を組み合わせることも可能です (たとえば、1-3 5

7-8 9)。

使用例

例1 構成されたすべてのプロセッサに関する情報を詳細モードで表示する

次の例では、構成されたすべてプロセッサに関する情報を詳細モードで表示します。

```
psrinfo -v
```

例2 プロセッサがオンラインであるかどうかを判定する

次の例では、シェルスクリプトで psrinfo を使用して、プロセッサがオンラインであるかどうかを判定します。

```
if [ "`psrinfo -s 3 2> /dev/null`" -eq 1 ]
then
    echo "processor 3 is up"
fi
```

例3 システム内の物理プロセッサに関する情報を表示する

追加引数を使用しない場合、-p オプションは、システム内の物理プロセッサの数を示す単一の整数を表示します。

```
> psrinfo -p
                8
```

psrinfo は、コマンド行引数 (プロセッサ ID) も受け入れます。

```
> psrinfo -p 0 512   # IDs 0 and 512 exist on the
1                    # same physical processor

> psrinfo -p 0 1     # IDs 0 and 1 exist on different
2                    # physical processors
```

この例では、仮想プロセッサ 0 および 512 が同じ物理プロセッサ上に存在します。仮想プロセッサ 0 および 1 は存在しません。これはこの例にかぎったものであり、一般的な規則ではありません。

終了ステータス

次の終了値が返されます。

0 正常終了。

>0 エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目	psradm(1M), p_online(2), processor_info(2), attributes(5)
診断	psrinfo: processor 9: Invalid argument　指定されたプロセッサが存在しません。

名前

psrset – creation and management of processor sets

形式

```
psrset -a [-F] processor_set_id processor_id...
psrset -b processor_set_id pid [/lwpid]...
psrset -c [-F] [processor_id]...
psrset -d processor_set_id...
psrset -e processor_set_id command [argument(s)]
psrset -f processor_set_id
psrset [-i] [processor_set_id]...
psrset -n processor_set_id
psrset -p [processor_id]...
psrset [-q] [pid [/lwpid]]...
psrset -Q [processor_set_id]...
psrset -r [-F] processor_id...
psrset -u pid [/lwpid]...
psrset -U [processor_set_id]...
```

機能説明

The `psrset` utility controls the management of processor sets. Processor sets allow the binding of processes or LWPs to groups of processors, rather than just a single processor. Processors assigned to processor sets can run only LWPs that have been bound to that processor set.

This command cannot be used to modify processor disposition when pools are enabled. Use `pooladm(1M)` and `poolcfg(1M)` to modify processor set configuration through the resource pools facility.

オプション

The following options are supported:

`-a` Assign the specified processors to the specified processor set. With the additional `-F` option, all LWPs bound to the specified processors will be unbound prior to changing processor sets.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-b` Bind all or a subset of the LWPs of the specified processes to the specified processor set.

LWPs bound to a processor set are restricted to run only on the processors in that set. Processes can only be bound to non-empty processor sets, that is, processor sets that have had processors assigned to them.

Bindings are inherited, so new LWPs and processes created by a bound LWP have the same binding. Binding an interactive shell to a processor, for example, binds all commands executed by the shell.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-c` Create a new processor set and displays the new processor set ID. With the additional `-F` option, all LWPs bound to the specified processors will be unbound prior to assigning them to the processor set being created.

If a list of processors is given, it also attempts to assign those processors to the processor set. If this succeeds, the processors are idle until LWPs are bound to the processor set. This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

Only a limited number of processor sets can be active (created and not destroyed) at a given time. This limit is always be greater than the number of processors in the system. If the `-c` option is used when the maximum number of processor sets is already active, the command fails.

The following format is used for the first line of output of the `-c` option when the `LC_MESSAGES` locale category specifies the "C" locale. In other locales, the strings `created`, `processor`, and `set` can be replaced with more appropriate strings corresponding to the locale.

`"created processor set %d\n"` *processor set ID*

`-d` Remove the specified processor set, releasing all processors and processes associated with it.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-e` Execute a command (with optional arguments) in the specified processor set.

The command process and any child processes are executed only by processors in the processor set.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-f` Disables interrupts for all processors within the specified processor set. See `psradm(1M)`.

If some processors in the set cannot have their interrupts disabled, the other processors still have their interrupts disabled, and the command reports an error and return non-zero exit status.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-F` Forces the specified processor set operation by unbinding all threads bound to the specified processor. Only the `-a` or the `-r` option can be used in combination with this option. Administrators are encouraged to use the `-Q` option for `pbind(1M)` to find out which threads will be affected by such operation.

`-i` Display a list of processors assigned to each named processor set. If no argument is given, a list of all processor sets and the processors assigned to them is displayed. This is also the default operation if the `psrset` command is not given an option.

`-n` Enable interrupts for all processors within the specified processor set. See `psradm(1M)`.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-p` Display the processor set assignments for the specified list of processors. If no argument is given, the processor set assignments for all processors in the system is given.

`-q` Display the processor set bindings of the specified processes or of all processes. If a process is composed of multiple LWPs which have different bindings and the LWPs are not explicitly specified, the bindings of only one of the bound LWPs is displayed. The bindings of a subset of LWPs can be displayed by appending "/lwpids" to the process IDs. Multiple LWPs may be selected using "-" and "," delimiters. See EXAMPLES.

`-Q` Display the LWPs bound to the specified list of processor sets, or all LWPs with processor set bindings.

`-r` Remove a list of processors from their current processor sets. Processors that are removed return to the general pool of processors.

Processors with LWPs bound to them using `pbind(1M)` can be assigned to or removed from processor sets using the `-F` option.

This option is restricted to users with the `PRIV_SYS_RES_CONFIG` privilege.

`-u` Remove the processor set bindings of a subset or all the LWPs of the specified processes, allowing them to be executed on any on-line processor if they are not bound to individual processors through `pbind`.

Users with the `PRIV_SYS_RES_CONFIG` privilege can unbind any process or LWP from any active processor set. Other users can unbind processes and LWPs from processor sets that do not have the `PSET_NOESCAPE` attribute set. In addition, the user must have permission to control the affected processes; the real or effective user ID of the user must match the real or saved user ID of the target processes.

`-U` Removes the bindings of all LWPs bound to the specified list of processor sets, or to any processor set if no argument is specified.

オペランド

The following operands are supported:

pid Specify *pid* as a process ID.

lwpid The set of LWPIDs of the specified process to be controlled or queried. The syntax for selecting LWP IDs is as follows:

```
2,3,4-8       LWP IDs 2, 3, and 4 through 8
-4            LWPs whose IDs are 4 or below
4-            LWPs whose IDs are 4 or above
```

processor_id Specify *processor_id* as an individual processor number (for example, `3`), multiple processor numbers separated by spaces (for example, `1 2 3`), or a range of processor numbers (for example, `1-4`). It is also possible to combine ranges and (individual or multiple) *processor_id*s (for example, `1-3 5 7-8 9`).

processor_set_id Specify *processor_set_id* as a processor set ID.

終了ステータス

The following exit values are returned:

`0` Successful completion.

non-0 An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`pbind(1M)`, `pooladm(1M)`, `poolcfg(1M)`, `psradm(1M)`, `psrinfo(1M)`, `processor_bind(2)`, `processor_info(2)`, `pset_bind(2)`, `pset_create(2)`, `pset_info(2)`, `sysconf(3C)`, `libpool(3LIB)`, `attributes(5)`, `privileges(5)`

診断

The following output indicates that the specified process did not exist or has exited:

```
psrset: cannot query pid 31: No such process
```

The following output indicates that the user does not have permission to bind the process:

```
psrset: cannot bind pid 31: Not owner
```

The following output indicates that the user does not have permission to assign the processor:

```
psrset: cannot assign processor 4: Not owner
```

The following output indicates that the specified processor is not on-line, or the specified processor does not exist.

```
psrset: cannot assign processor 8: Invalid argument
```

The following output indicates that an LWP in the specified process is bound to a processor and cannot be bound to a processor set that does not include that processor:

```
psrset: cannot bind pid 67: Device busy
```

The following output indicates that the specified processor could not be added to the processor set. This can be due to bound LWPs on that processor, or because that processor cannot be combined in the same processor set with other processors in that set, or because the processor is the last one in its current processor set:

```
psrset: cannot assign processor 7: Device busy
```

The following output indicates that the specified processor set does not exist:

```
psrset: cannot execute in processor set 8: Invalid argument
```

The following output indicates that the maximum number of processor sets allowed in the system is already active:

```
psrset: cannot create processor set: Not enough space
```

The following output indicates that the pools facility is active.

```
psrset: cannot assign processor 7: Operation not supported
psrset: cannot bind pid 31: Operation not supported
psrset: cannot bind pid 31: Operation not supported
psrset: could not create processor set: Operation not supported
psrset: could not remove processor set 1: Operation not supported
psrset: cannot exec in processor set 1: Operation not supported
psrset: cannot remove processor 7: Operation not supported
psrset: cannot unbind pid 31: Operation not supported
```

名前

pwck, grpck – password/group file checkers

形式

/usr/sbin/pwck [*filename*]

/usr/sbin/grpck [*filename*]

機能説明

pwck scans the password file and notes any inconsistencies. The checks include validation of the number of fields, login name, user ID, group ID, and whether the login directory and the program-to-use-as-shell exist. The default password file is /etc/passwd.

grpck verifies all entries in the group file. This verification includes a check of the number of fields, group name, group ID, whether any login names belong to more than NGROUPS_MAX groups, and that all login names appear in the password file. grpck also issues a warning if it finds an entry (a single line) in the group file longer than 2047 characters. Such an entry causes group maintenance commands, such as groupdel(1M) and groupmod(1M), to fail.

The default group file is /etc/group.

All messages regarding inconsistent entries are placed on the stderr stream.

ファイル

- /etc/group
- /etc/passwd

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

groupdel(1M), groupmod(1M), getpwent(3C), group(4), passwd(4), attributes(5)

診断

Group entries in /etc/group with no login names are flagged.

Group file '*filename*' is empty
: The /etc/passwd or /etc/group file is an empty file.

cannot open file *filename*: No such file or directory
: The /etc/passwd or /etc/group file does not exist.

注意事項

If no filename argument is given, grpck checks the local group file, /etc/group, and also makes sure that all login names encountered in the checked group file are known to the system getpwent(3C) routine. This means that the login names may be supplied by a network name service.

名前

pwconv – installs and updates /etc/shadow with information from /etc/passwd

形式

```
pwconv
```

機能説明

The pwconv command creates and updates /etc/shadow with information from /etc/passwd.

pwconv relies on a special value of 'x' in the password field of /etc/passwd. This value of 'x' indicates that the password for the user is already in /etc/shadow and should not be modified.

If the /etc/shadow file does not exist, this command will create /etc/shadow with information from /etc/passwd. The command populates /etc/shadow with the user's login name, password, and password aging information. If password aging information does not exist in /etc/passwd for a given user, none will be added to /etc/shadow. However, the last changed information will always be updated.

If the /etc/shadow file does exist, the following tasks will be performed:

Entries that are in the /etc/passwd file and not in the /etc/shadow file will be added to the /etc/shadow file.

Entries that are in the /etc/shadow file and not in the /etc/passwd file will be removed from /etc/shadow.

Password attributes (for example, password and aging information) that exist in an /etc/passwd entry will be moved to the corresponding entry in /etc/shadow.

The pwconv command can only be used by the super-user.

ファイル

- /etc/opasswd
- /etc/oshadow
- /etc/passwd
- /etc/shadow

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

passwd(1), usermod(1M), passwd(4), attributes(5)

診断

pwconv exits with one of the following values:

0 SUCCESS.

1 Permission denied.

2 Invalid command syntax.

3 Unexpected failure. Conversion not done.

4 Unexpected failure. Password file(s) missing.

5 Password file(s) busy. Try again later.

6 Bad entry in `/etc/shadow` file.

名前

quot – summarize file system ownership

形式

`quot [-acfhnv] filesystem...`

`quot -a [-cfhnv]`

機能説明

`quot` displays the number of blocks (1024 bytes) in the named *filesystem* (one or more) currently owned by each user. There is a limit of 2048 blocks. Files larger than this will be counted as a 2048 block file, but the total block count will be correct.

オプション

The following options are supported:

`-a` Generate a report for all mounted file systems.

`-c` Display three columns giving a file size in blocks, the number of files of that size, and a cumulative total of blocks containing files of that size or a smaller size.

`-f` Display three columns giving, for each user, the number of blocks owned, the count of number of files, and the user name. This option is incompatible with the `-c` and `-v` options.

`-h` Estimate the number of blocks in the file. This does not account for files with holes in them.

`-n` Attach names to the list of files read from standard input. `quot -n` cannot be used alone, because it expects data from standard input. For example, the pipeline

```
ncheck myfilesystem | sort +0n | quot -n myfilesystem
```

will produce a list of all files and their owners. This option is incompatible with all other options.

`-v` In addition to the default output, display three columns containing the number of blocks not accessed in the last 30, 60, and 90 days.

オペランド

filesystem mount-point of the filesystem(s) being checked

使用法

See `largefile(5)` for the description of the behavior of `quot` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

終了ステータス

`0` Successful operation.

`32` Error condition (bad or missing argument, bad path, or other error).

ファイル

`/etc/mnttab` Lists mounted file systems.

`/etc/passwd` Used to obtain user names

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

du(1), mnttab(4), passwd(4), attributes(5), largefile(5)

注意事項

This command can only be used by the super-user.

名前

quota – display a user's ufs or zfs file system disk quota and usage

形式

`quota [-v] [`*`username`*`]`

機能説明

`quota` displays users' UFS or ZFS disk usage and limits. Only the super-user may use the optional *username* argument to view the limits of other users.

`quota` without options only display warnings about mounted file systems where usage is over quota. Remotely mounted file systems which do not have quotas turned on are ignored.

username can be the numeric UID of a user.

オプション

`-v` Display user's quota on all mounted file systems where quotas exist.

使用法

See `largefile(5)` for the description of the behavior of `quota` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/mnttab` list of currently mounted filesystems

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`edquota(1M)`, `quotaon(1M)`, `quotacheck(1M)`, `repquota(1M)`, `rquotad(1M)`, `attributes(5)`, `largefile(5)`, `zones(5)`

注意事項

`quota` displays quotas for NFS mounted UFS- or ZFS-based file systems if the `rquotad` daemon is running. See `rquotad(1M)`. In a `zones(5)` environment, `quota` displays quotas only for the zone in which it is invoked.

`quota` can display entries for the same file system multiple times for multiple mount points. For example,

```
# quota -v user1
```

might display identical quota information for `user1` at the mount points `/home/user1`, `/home/user2`, and `/home/user`, if all three mount points are mounted from the same file system with quotas turned on.

名前

quotacheck – ufs file system quota consistency checker

形式

```
quotacheck [-fp] [-v] filesystem...
quotacheck -a [-fpv]
```

機能説明

`quotacheck` examines each mounted ufs file system, builds a table of current disk usage, and compares this table against the information stored in the file system's disk quota file. If any inconsistencies are detected, both the quota file and the current system copy of the incorrect quotas are updated.

filesystem is either a file system mount point or the block device on which the file system resides.

`quotacheck` expects each file system to be checked to have a quota file named `quotas` in the root directory. If none is present, `quotacheck` will not check the file system.

`quotacheck` accesses the character special device in calculating the actual disk usage for each user. Thus, the file systems that are checked should be quiescent while `quotacheck` is running.

オプション

The following options are supported:

- `-a` Check the file systems which `/etc/mnttab` indicates are ufs file systems. These file systems must be read-write mounted with disk quotas enabled, and have an `rq` entry in the `mntopts` field in `/etc/vfstab`.
- `-f` Force check on file systems with logging enabled. Use in combination with the `-p` option.
- `-p` Check quotas of file systems in parallel. For file systems with logging enabled, no check is performed unless the `-f` option is also specified.
- `-v` Indicate the calculated disk quotas for each user on a particular file system. `quotacheck` normally reports only those quotas modified.

使用法

See `largefile(5)` for the description of the behavior of `quotacheck` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/mnttab`	Mounted file systems
`/etc/vfstab`	List of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

edquota(1M), quota(1M), quotaon(1M), repquota(1M), attributes(5), largefile(5), quotactl(7I), mount_ufs(1M)

名前

quotaon, quotaoff – turn ufs file system quotas on and off

形式

```
quotaon [-v] filesystem...
quotaon -a [-v]
quotaoff [-v] filesystem...
quotaoff -a [-v]
```

機能説明

`quotaon` turns on disk quotas for one or more ufs file systems.

Before a file system may have quotas enabled, a file named `quotas`, owned by root, must exist in the root directory of the file system. See `edquota(1M)` for details on how to modify the contents of this file.

`quotaoff` turns off disk quotas for one or more ufs file systems.

The file systems specified must already be mounted.

These commands update the `mntopts` field of the appropriate entries in `/etc/mnttab` to indicate when quotas are on or off for each file system. If quotas are on, the string `quota` will be added to `mntopts`; if quotas are off, the `quota` string is not present.

filesystem must be either the mount point of a file system, or the block device on which the file system resides.

オプション

quotaon

`-a` This option is normally used at boot time to enable quotas. It applies only to those file systems in `/etc/vfstab` which have "rq" in the `mntopts` field, are currently mounted ' 'rw", and have a `quotas` file in the root directory.

`-v` Display a message for each file system after quotas are turned on.

quotaoff

`-a` Force all file systems in `/etc/mnttab` to have their quotas disabled.

`-v` Display a message for each file system affected.

使用法

See `largefile(5)` for the description of the behavior of `quotaon` and `quotaoff` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

`/etc/mnttab` mounted file systems

`/etc/vfstab` list of default parameters for each file system

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

edquota(1M), quota(1M), quotacheck(1M), repquota(1M), mnttab(4), vfstab(4), attributes(5), largefile(5), quotactl(7I)

名前

raidctl – RAID ハードウェアユーティリティー

形式

```
raidctl -C "disks" [-r raid_level] [-z capacity] [-s stripe_size] [-f]
     controller

raidctl -d [-f] volume

raidctl -F filename [-f] controller...

raidctl -a {set | unset} -g disk {volume | controller}

raidctl -p "param=value" [-f] volume

raidctl -c [-f] [-r raid_level] disk1 disk2 [disk3...]

raidctl -l -g disk controller

raidctl -l volume

raidctl -l controller...

raidctl [-l]

raidctl -S [volume | controller]

raidctl -S -g disk controller

raidctl -h
```

機能説明

raidctl ユーティリティーはハードウェア RAID 構成ツールであり、RAID ボリュームを作成、削除、または表示する CLI (コマンド行インタフェース) をエンドユーザーに提供することで、さまざまな RAID コントローラをサポートしています。このユーティリティーはまた、ボリュームのプロパティーの設定、ボリュームやコントローラへのホットスペア (HSP) ディスクの割り当て、および RAID コントローラのファームウェア/fcode/BIOS の更新にも使用できます。

raidctl ユーティリティーには、基礎となるファイルシステムのアクセス権で制御される特権が必要です。特権を持つユーザーだけが RAID システム構成を操作できます。特権を持たないユーザーが raidctl を実行しようとすると、終了ステータス 1 が返されてコマンドは失敗します。

raidctl ユーティリティーは、このマニュアルページで説明しているように、完全な機能を備えた RAID コントローラを管理するための広範にわたるコマンド行オプションのセットを定義しています。ただし、特定のオプションのサポートは次の 2 つの要素に依存しています。

- ソフトウェアドライバの存在
- RAID デバイスのファームウェアレベル

ソフトウェアドライバへの依存は、raidctl の設計によるものです。このユーティリティーは、各種ドライバ向けのプラグインモジュールの挿入を可能にする共通ライブラリに基づいて構築されます。現時点では、Solaris オペレーティングシステムには mpt ドライバ向けのプラグインが付属しています。このプラグインは、すべての raidctl オプションをサポートしているわけではありません。特定の

ストレージデバイスでは、オプションはそのデバイスのファームウェアレベルによってさらに制限される場合があります。

さまざまな raidctl オプションのサポートレベルを raidctl によって決めることはできません。ユーザーは、RAID コントローラやハードウェアプラットフォームのドキュメントに頼る必要があります。

現時点では、raidctl は次の RAID コントローラに対してある程度のサポートを提供しています。

- LSI1020 SCSI HBA
- LSI1030 SCSI HBA
- LSI1064 SAS HBA
- LSI1068 SAS HBA

X86-32/64 および SPARC プラットフォームでは、上記のすべての HBA は mpt ドライバによって維持されます。

オプション

次のオプションがサポートされています。

-C *"disks"* [-r *raid_level*] [-z *capacity*] [-s *stripe_size*] [-f] controller
指定したディスクを使用して RAID ボリュームを作成します。

このオプションを使用して RAID ボリュームを作成すると、新しく作成されたボリュームの識別情報が自動的に生成され、raidctl はそれをユーザーに報告します。

このオプションによって指定される引数には、作成されるボリュームの構成に使われる要素が含まれます。要素にはディスクまたはサブボリュームのいずれかを設定できます。ディスクはスペースで区切り、サブボリュームはディスクのセットを丸括弧でグループ化します。すべてのディスクは「C.ID.L」の形式にしてください (たとえば「0.1.2」は、チャネル 0 の物理ディスクであり、ターゲット ID は 1、論理ユニット番号は 2 であることを表す)。この引数は、(-r オプションを省略した場合でも) -r オプションで指定した RAID レベルと一致している必要があります。つまり、指定できる引数は次のように限定されます。

RAID 0 の場合
少なくとも 2 つのディスク

RAID 1 の場合
2 つのディスクのみ

RAID 1E の場合
少なくとも 3 つのディスク

RAID 5 の場合
少なくとも 3 つのディスク

RAID 10 の場合
少なくとも 2 つのサブボリューム。各サブボリュームは 2 つのディスクで構成されている必要があります

RAID 50 の場合
少なくとも 2 つのサブボリューム。各サブボリュームは 3 つ以上のディスクで構成されている必要があり、各サブボリューム内のディスク容量は同じであるべきです

たとえば、「0.0.0 0.1.0」の形式は、指定された 2 つのディスクが RAID ボリューム (RAID 0 または RAID 1 ボリュームのいずれか) を構成することを意味します。「(0.0.0 0.1.0)(0.2.0 0.3.0)」は、最初の 2 つのディスクと最後の 2 つのディスクが 2 つのサブボリュームを構成し、これら 2 つのサブボリュームが RAID 10 ボリュームを構成することを意味します。その他の例については、「使用例」の節を参照してください。

`-r` オプションは、作成されるボリュームの RAID レベルを指定します。指定できるレベルは 0、1、1E、5、10、50 です。このオプションを省略すると、`raidctl` はデフォルトで RAID 1 ボリュームを作成します。

`-z` オプションは、作成されるボリュームの容量を指定します。単位として、テラバイト、ギガバイト、またはメガバイト (たとえば 2t、10g、20m など) を使用できます。このオプションを省略すると、`raidctl` は指定されたディスクで作成できるボリュームの最大容量を計算し、この値を使用してボリュームを作成します。

`-s` オプションは、作成されるボリュームのストライプサイズを指定します。指定できる値は 512、1k、2k、4k、8k、16k、32k、64k、または 128k です。このオプションを省略すると、`raidctl` はボリュームに適した値 (たとえば 64k) を選択します。

たとえば LSI1020、LSI1030、SAS1064、または SAS1068 HBA において、場合によっては RAID ボリュームの作成が特定のディスク上のデータ消失を引き起こすことがあり、`raidctl` はユーザーに対してボリューム作成の確認を要求します。ユーザーへの確認要求なしでボリューム作成を強制させるには、`-f` オプションを使用します。

controller 引数は、指定したディスクがどの RAID コントローラに属するかを特定するために使用します。`-l` オプションを使用すれば、コントローラの ID 番号を一覧表示できます。

`-d` [`-f`] *volume*
volume として指定された RAID ボリュームを削除します。ボリュームは標準的な形式 (たとえば `c0t0d0`) で指定します。

ボリュームを削除すると、すべてのデータが失われます。このため、`-f` オプションを指定しないかぎり、`raidctl` はボリュームを削除する前にユーザーに対して確認を要求します。

LSI1020、LSI1030、SAS1064、または SAS1068 HBA から RAID 1 ボリュームを削除すると、主ディスクと二次ディスクが「分離」します。ボリュームが SYNCING 状態にあった場合、一次側にはデータが含まれていますが、二次側には含まれていません。ボリュームの状態が OPTIMAL であった場合、両方のディスクにデータの完全なイメージが含まれています。

-F *filename* [-f] *controller*…
指定したコントローラで動作しているファームウェアを更新します。-f オプションを指定しないかぎり、raidctl ユーティリティーはユーザーに対してこのアクションの確認を要求します。

-a {set | unset} -g *disk* {volume | controller}
ボリュームを指定した場合、-a で指定した値によって、raidctl はディスクをボリューム専用のローカルホットスペアディスクとして設定するか、その設定を解除します。コントローラを指定した場合、raidctl はディスクをグローバルなホットスペアディスクとして設定するか、その設定を解除します。

-p *"param=value"* [-f] *volume*
指定した RAID ボリュームのプロパティー値を変更します。このオプションは、キャッシュ書き込みポリシーの変更やボリュームのアクティブ化に使用できます。キャッシュ書き込みポリシーの変更時は、*param* を文字列 wp (SET_WR_POLICY) にするようにしてください。*value* には on または off を設定できます。ボリュームのアクティブ化に使用するときは、*param* を state にし、*value* を activate にするようにしてください。

RAID ボリュームのプロパティーを変更すると RAID コントローラの内部動作に影響する可能性があるため、-f オプションを指定しないかぎり、raidctl は変更を適用する前にユーザーに対して確認を要求します。

-c [-f] [-r *raid_level*] *disk1 disk2* [*disk3*...]
指定したディスクを使用してボリュームを作成します。これは、似た機能を持つ -C オプションの代替手段です。このオプションは互換性のために残されていますが、LSI1020、LSI1030、SAS1064、および SAS1068 HBA で RAID 0、RAID 1、または RAID 1E ボリュームを作成することしかできません。ほかの HBA では、ユーザーは -C オプションのみを使用できます。

-r オプションは、ターゲットボリュームの RAID レベルを指定するために使用できます。-r オプションを省略すると、raidctl は RAID 1 ボリュームを作成します。

ディスクは Solaris の標準的な形式 (たとえば c0t0d0) で指定する必要があります。

このオプションで RAID 1 ボリュームを作成すると、disk2 の内容が disk1 の内容に置き換わります。

ユーザーがこのオプションで RAID ボリュームを作成すると、RAID ボリュームは disk1 の識別情報を引き受けます。ほかのディスクは見えなくなり、RAID ボリュームは 1 つのディスクとして表示されます。

このオプションによるボリュームの作成は、デフォルトで対話形式で進められます。ボリュームを作成するには、ユーザーはプロンプトに明示的に答える必要があります。ユーザーへの確認要求なしでボリューム作成を強制させるには、`-f` オプションを使用します。

`-l` `-g` *disk controller*
指定したコントローラの、指定したディスクに関する情報を表示します。出力には次の情報が含まれます。

Disk
ディスクを `C.ID.L` の形式で表示します。

Vendor
ベンダー ID の文字列を表示します。

Product
製品 ID の文字列を表示します。

Capacity
ディスクの合計容量を表示します。

Status
ディスクの現在のステータスを表示します。ステータスは、「`GOOD`」(正常動作中)、「`FAILED`」(機能していない)、または「`MISSING`」(ディスクが存在しない) のいずれかになります。

HSP
ディスクがグローバルホットスペアディスク、ローカルホットスペアディスク、または通常のものとして設定されているかを示します。ローカルホットスペアディスクの場合、このディスクが割り当てられているすべてのボリュームが表示されます。

GUID
指定されたディスクの GUID 文字列。これは追加のデータであり、場合によっては利用できないことがあります。

`-l` *volume*
指定したボリュームに関する情報を表示します。出力には次の情報が含まれます。

Volume
ボリュームを標準的な形式で表示します。

Sub
指定したボリュームが RAID 10 または RAID 50 ボリュームの場合、サブボリュームを表示します。

Disk
指定したボリュームを構成するディスクをすべて表示します。

Stripe Size
ボリュームのストライプサイズを表示します。

Status
指定したボリューム、または指定したボリュームを構成するサブボリュームやディスクのステータスを表示します。アクティブでないボリュームの場合、ステータスは `INACTIVE` になるはずです。それ以外の場合、ステータスは `OPTIMAL` (正常動作中)、`DEGRADED` (機能制限付きで動作中)、`FAILED` (機能していない)、または `SYNC` (ディスクが同期中) のいずれかになります。ディスクの場合、ステータスは `GOOD`、`FAILED`、または `MISSING` のいずれかになります。

Cache
キャッシュの入出力書き込み動作が有効になっているかどうかを示します。キャッシュは「`ON`」または「`OFF`」のいずれかです。

RAID level
RAID レベルを表示します。RAID レベルは 0、1、1E、5、10、または 50 のいずれかです。

`-l` *controller* ...
指定したコントローラに関する情報を表示します。出力には次の情報が含まれます。

Controller
RAID コントローラの ID 番号を表示します。

Type
RAID コントローラの製品タイプを表示します。

fw_version
コントローラのファームウェアのバージョンを表示します。

[`-l`]
`raidctl` ユーティリティーが操作できる、RAID に関連するすべてのオブジェクトを一覧表示します。これには、使用可能なすべての RAID コントローラ、RAID ボリューム、および物理ディスクが含まれます。`-l` オプションは省略できます。

出力には次の情報が含まれます。

Controller
RAID コントローラの ID 番号を表示します。

Volume
論理 RAID ボリューム名を表示します。

Disk
RAID ディスクを `C.ID.L` の形式で表示します。

-S [volume | controller]
使用可能なすべての RAID デバイス、RAID コントローラ、ボリューム、およびディスクを含む、RAID 構成情報のスナップショットを取得します。

出力の各行には、RAID デバイスとその関連情報がスペースで区切られて表示されます。すべてのボリュームとディスクは、最後に指定したコントローラに属します。

出力には次の情報が一覧表示されます。

Controller
コントローラの ID 番号と、コントローラのタイプを表す文字列 (二重引用符に囲まれている) が表示されます。

Volume
RAID ボリューム名、コンポーネントのディスクの数、コンポーネントのディスクの C.ID.L 形式、RAID レベル、およびステータスを表示します。ステータスは「OPTIMAL」、「DEGRADED」、「FAILED」、または「SYNCING」のいずれかです。

Disk
ディスクの C.ID.L 形式とステータスを表示します。ステータスは「GOOD」、「FAILED」、または「HSP」(ディスクが予備用ディスクとして設定されている) のいずれかです。

ボリュームまたはコントローラを指定すると、指定したボリュームまたはコントローラの情報のスナップショットしか取得されません。

-S -g *disk controller*
指定したディスクの情報のスナップショットを取得します。

-h
使用法についての文字列を出力します。

使用例

例 1 RAID 構成の作成

次のコマンドは、コントローラ 0 に 10G の RAID 0 ボリュームを作成し、ストライプサイズは 64k に設定されます。

```
# raidctl -C "0.0.0 0.2.0" -r 0 -z 10g -s 64k 0
```

次のコマンドは、コントローラ 2 に RAID 1 ボリュームを作成します。

```
# raidctl -C "0.0.0 1.1.0" -r 1 2
```

次のコマンドは、コントローラ 2 に RAID 5 ボリュームを作成します。

```
# raidctl -C "0.0.0 0.1.0 0.2.0" -r 5 2
```

次のコマンドは、コントローラ 0 に RAID 10 ボリュームを作成します。

```
# raidctl -C "(0.0.0 0.1.0)(0.2.0 0.3.0)" -r 10 0
```

例 1　RAID 構成の作成　　*(続き)*

次のコマンドは、コントローラ 0 に RAID 50 ボリュームを作成します。

```
# raidctl -C "(0.0.0 0.1.0 0.2.0)(0.3.0 0.4.0 0.5.0)" -r 50 0
```

例 2　RAID 構成の表示

次のコマンドは、使用可能なすべてのコントローラ、ボリューム、およびディスクを表示します。

```
# raidctl -l

Controller: 0
Controller: 2
         Volume:c2t0d0
         Disk: 0.0.0
         Disk: 0.1.0
         Disk: 0.2.0
         Disk: 0.3.0(HSP)
```

次のコマンドは、コントローラ 2 に関する情報を表示します。

```
# raidctl -l 2

Controller      Type            Fw_version
----------------------------------------------------------------
c2              LSI 1030        1.03.39.00
```

次のコマンドは、指定したボリュームに関する情報を表示します。

```
# raidctl -l c2t0d0

Volume                  Size    Stripe  Status  Cache   RAID
         Sub                     Size                    Level
                Disk
----------------------------------------------------------------
c2t0d0                 10240M  64K     OPTIMAL ON      RAID5
                0.0.0   5120M           GOOD
                0.1.0   5120M           GOOD
                0.2.0   5120M           GOOD
```

次のコマンドは、コントローラ 0 のディスク 0.0.0 に関する情報を表示します。

```
# raidctl -l -g 0.0.0  0

Disk    Vendor  Product         Firmware Capacity        Status  HSP
-----------------------------------------------------------------------
0.0.0   HITACHI H101473SCSUN72G SQ02     68.3G           GOOD    N/A
GUID:2000000cca02536c
```

例 3　RAID 構成の削除

次のコマンドは、ボリュームを削除します。

```
# raidctl -d c0t0d0
```

例 4　コントローラのフラッシュイメージの更新

次のコマンドは、コントローラ 0 のフラッシュイメージを更新します。

```
# raidctl -F lsi_image.fw 0
```

例 5　ホットスペアディスクの設定または設定解除

次のコマンドは、コントローラ 2 のディスク 0.3.0 をグローバルホットスペアディスクとして設定します。

```
# raidctl -a set -g 0.3.0 2
```

次のコマンドは、コントローラ 2 のディスク 0.3.0 を、ボリューム c2t0d0 に対するローカルホットスペアディスクとして設定します。

```
# raidctl -a set -g 0.3.0 c2t0d0
```

次のコマンドは、コントローラ 2 のディスク 0.3.0 を、グローバルホットスペアディスクから通常のディスクへ変換します。

```
# raidctl -a unset -g 0.3.0 2
```

次のコマンドは、ディスク 0.3.0 のボリューム c2t0d0 に対するローカルホットスペアディスクの設定を解除します。

```
# raidctl -a unset -g 0.3.0 c2t0d0
```

例 6　ボリュームプロパティーの設定

次のコマンドは、ボリュームの書き込みポリシーを「`off`」に設定します。

```
# raidctl -a set -p "wp=off" c0t0d0
```

例 7　-c オプションによるボリュームの作成

次のコマンドは、RAID 1 ボリュームを作成します。

```
# raidctl -c c0t0d0 c0t1d0
```

次のコマンドは、RAID 0 ボリュームを作成します。

```
# raidctl -c -r 0 c0t1d0 c0t2d0 c0t3d0
```

例 8　RAID 構成のスナップショットの取得

次のコマンドは、すべての RAID デバイスのスナップショットを取得します。

例 8　RAID 構成のスナップショットの取得　　(続き)

```
# # raidctl -S

1 "LSI 1030"
c1t1d0 2 0.2.0 0.3.0 1 DEGRADED
0.2.0 GOOD
0.3.0 FAILED
```

次のコマンドは、ボリューム c1t0d0 についてのスナップショットを取得します。

```
# raidctl -S c1t0d0

c1t0d0 2 0.0.0 0.1.0 1 OPTIMAL
```

次のコマンドは、コントローラ 1 のディスク 0.1.0 についてのスナップショットを取得します。

```
# raidctl -S -g 0.1.0 1

0.1.0 GOOD
```

終了ステータス

次の終了値が返されます。

0
　正常終了。

1
　コマンド行入力が不正、またはアクセスが拒否されました。

2
　要求操作は失敗しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

attributes(5), mpt(7D)

『『Oracle Solaris の管理: 一般的なタスク』』

警告

Solaris マルチパス入出力機能 (MPxIO としても知られる) を使用する場合は、内部 SAS ディスク上に RAID ボリュームを作成しないでください。Solaris マルチパス環境で新しい RAID ボリュームを作成すると、既存デバイスの GUID と一致しない新しい GUID が root デバイスに与えられます。これによって /etc/vfstab 内の root デバイスのエントリが一致しなくなり、ブート障害を引き起こします。

注意事項

`-z` オプションは、`mpt` ドライバおよび LSI RAID コントローラを使用するシステムではサポートされません。

名前

ramdiskadm – administer ramdisk pseudo device

形式

```
/usr/sbin/ramdiskadm -a name size [g | m | k | b]
/usr/sbin/ramdiskadm -d name
/usr/sbin/ramdiskadm
```

機能説明

The `ramdiskadm` command administers `ramdisk(7D)`, the ramdisk driver. Use `ramdiskadm` to create a new named ramdisk device, delete an existing named ramdisk, or list information about existing ramdisks.

Ramdisks created using `ramdiskadm` are not persistent across reboots.

オプション

The following options are supported:

`-a` *name size*
: Create a ramdisk named *name* of size *size* and its corresponding block and character device nodes.

  name must be composed only of the characters a-z, A-Z, 0-9, _ (underbar), and - (hyphen), but it must not begin with a hyphen. It must be no more than 32 characters long. Ramdisk names must be unique.

  The size can be a decimal number, or, when prefixed with `0x`, a hexadecimal number, and can specify the size in bytes (no suffix), 512-byte blocks (suffix `b`), kilobytes (suffix `k`), megabytes (suffix `m`) or gigabytes (suffix `g`). The size of the ramdisk actually created might be larger than that specified, depending on the hardware implementation.

  If the named ramdisk is successfully created, its block device path is printed on standard out.

`-d` *name*
: Delete an existing ramdisk of the name *name*. This command succeeds only when the named ramdisk is not open. The associated memory is freed and the device nodes are removed.

  You can delete only ramdisks created using `ramdiskadm`. It is not possible to delete a ramdisk that was created during the boot process.

Without options, `ramdiskadm` lists any existing ramdisks, their sizes (in decimal), and whether they can be removed by `ramdiskadm` (see the description of the `-d` option, above).

使用例

例 1　Creating a 2MB Ramdisk Named mydisk

```
# ramdiskadm -a mydisk 2m
/dev/ramdisk/mydisk
```

例 2 Listing All Ramdisks

```
# ramdiskadm
Block Device                    Size  Removable
/dev/ramdisk/miniroot     134217728    No
/dev/ramdisk/certfs         1048576    No
/dev/ramdisk/mydisk         2097152    Yes
```

終了ステータス

ramdiskadm returns the following exit values:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

attributes(5), ramdisk(7D)

注意事項

The abilities of ramdiskadm and the privilege level of the person who uses the utility are controlled by the permissions of /dev/ramdiskctl. Read access allows query operations, for example, listing device information. Write access is required to do any state-changing operations, for example, creating or deleting ramdisks.

As shipped, /dev/ramdiskctl is owned by root, in group sys, and mode 0644, so all users can do query operations but only root can perform state-changing operations. An administrator can give write access to non-privileged users, allowing them to add or delete ramdisks. However, granting such ability entails considerable risk; such privileges should be given only to a trusted group.

名前

rcapadm – configure resource capping daemon

形式

```
rcapadm

rcapadm [ [-n] -E | -D]
     [-i interval=value,...,interval=value] [-c percent]
     [-z zonename -m maxvalue]
```

機能説明

The `rcapadm` command allows a user with the privileges described below to configure various attributes of the resource capping daemon. If used without arguments, `rcapadm` displays the current status of the resource capping daemon if it has been configured. See `rcapd(1M)` for more information.

In the current release of the Solaris operating environment, `rcapadm` is available to users with all privileges and to users who have the Process Management profile in their list of profiles. The System Administrator role includes the Process Management profile.

オプション

`-c` *percent*

Set the minimum physical memory utilization for memory cap enforcement. Caps will not be enforced until the physical memory available to processes is low. The *percent* value should be in the range 0 to 100. The minimum (and default) value is `0`, which means that memory caps are always enforced.

`-D`

Disable the resource capping daemon so that it will not be started when the system is booted. Also stop the resource capping daemon now, if the `-n` option is not specified and it is currently running.

`-E`

Enable the resource capping daemon so that it will be started each time the system is booted. Also start the resource capping daemon now, if the `-n` option is not specified and it is not currently running.

`-i` *interval=value,...,interval=value*

Set intervals for various periodic operations performed by `rcapd`. All intervals are specified in seconds. You can set the following intervals:

`scan`

The interval at which `rcapd` scans for new processes. The default scan interval is every 15 seconds. The minimum value is 1 second.

`sample`

The interval of process resident set size sampling. The default sample interval is every 5 seconds. The minimum value is 1 second.

`report` The interval at which various paging statistics are updated by `rcapd`, in seconds. These statistics can be viewed by using `rcapstat`(1SRM). The default reporting interval is every 5 seconds. When the interval is set to `0`, statistics will not be updated.

注 – Paging refers to the act of relocating portions of memory, called pages, to or from physical memory. `rcapd` pages out the most infrequently used pages.

`config` The reconfiguration interval, in seconds. At each reconfiguration event, `rcapd` checks its configuration file for updates, and scans the project databases for new project caps. The default reconfiguration interval is every 60 seconds. The minimum interval is `0`. When the interval is set to `0`, no periodic reconfiguration occurs, although the running daemon can still be reconfigured by sending it `SIGHUP`.

`-m` *maxvalue* Used in conjunction with the `-z` option. Specifies a value for `rcap.max-rss`, a dynamically-set cap on the usage of physical memory for the zone specified by `-z`. You can apply a scale (`K`, `M`, `G`, `T`) to the value you specify. `K` means kilobyte; `M`, megabyte; `G`, gigabyte; and `T`, terabyte. For example, `100M` is 100 megabytes.

To remove an existing cap, specify `0M`.

`-n` Do not affect the running state of the resource capping daemon when enabling or disabling it.

`-z` *zonename* Used in conjunction with the `-m` option. Specifies the zone for which you are dynamically specifying a cap on physical memory usage (using `-m`).

注 – To set a persistent cap on memory usage within a zone, use `zonecfg`(1M).

使用例

例 1 Configuring the Resource Capping Daemon with Immediate Enforcement

```
# rcapadm -E -i scan=15,sample=5,report=5,config=60 -c 0
```

例 2 Specifying a Resource Cap for a Zone

The command shown below specifies the maximum amount of memory that can be consumed by a specified zone. Note that this value lasts only until the next reboot. To set a persistent cap, use zonecfg(1M).

```
# rcapadm -z testzone -m 512M
```

終了ステータス

The following exit values are returned:

0 Successful completion. The modifications to the current configuration were valid and made successfully.

1 An error occurred. A fatal error occurred either in obtaining or modifying the resource capping configuration.

2 Invalid command-line options were specified.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/resource-mgmt/resource-caps
Interface Stability	Committed

The -z and -m options are committed interfaces.

関連項目

rcapstat(1), rcapd(1M), zonecfg(1M), project(4), attributes(5), zones(5)

"Physical Memory Control Using the Resource Capping Daemon" in 『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

名前

rcapd – resource cap enforcement daemon

形式

```
rcapd [-d]
```

機能説明

The rcapd daemon enforces resource caps on collections of processes. Per-project and per-zone physical memory caps are supported. For information about projects, see project(4). For zones information, see zones(5)

When the resident set size (RSS) of a collection of processes exceeds its cap, rcapd takes action and reduces the RSS of the collection.

The virtual memory system divides physical memory into segments known as pages. To read data from a file into memory, the virtual memory system reads in individual pages. To reduce resource consumption, the daemon can page out, or relocate, infrequently used pages to an area outside of physical memory.

In the project file, caps are defined for projects that have positive values for the following project attribute:

rcap.max-rss　The total amount of physical memory, in bytes, that is available to the project's member processes

See project(4) for a description of project attributes.

For a system with one or more zones, you can dynamically set the rcap.max-rss value for a zone with rcapadm(1M). To set a persistent cap on memory usage within a zone, you use zonecfg(1M).

You configure rcapd through the use of rcapadm(1M). The daemon can be monitored with rcapstat(1). Configuration changes are incorporated into rcapd by sending it SIGHUP (see kill(1)), or according to the configuration interval (see rcapadm(1M)).

オプション

The following option is supported:

-d　Enable debug mode. Messages are displayed on the invoking user's terminal.

使用例

例 1　Setting Resident Set Size Cap Attribute

The following line in the /etc/project database sets an RSS cap of 1073741824 bytes for a project named foo.

```
foo:100::foo,root::rcap.max-rss=10737418240
```

終了ステータス

The following exit values are returned:

0　Successful completion.

1　An error occurred.

2　Invalid command-line options were specified.

ファイル

`/etc/project` Project database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/resource-mgmt/resource-caps
Interface Stability	Committed

関連項目

`rcapstat(1)`, `svcs(1)`, `rcapadm(1M)`, `zonecfg(1M)`, `svcadm(1M)`, `project(4)`, `attributes(5)`, `smf(5)`, `zones(5)`

"Physical Memory Control Using the Resource Capping Daemon" in 『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

注意事項

If killed with `SIGKILL`, `rcapd` can leave processes in a stopped state. Use `SIGTERM` to cause `rcapd` to terminate properly.

A collection's RSS can exceed its cap for some time before the cap is enforced, even if sufficient pageable memory is available. This period of time can be reduced by shortening the RSS sampling interval with `rcapadm`.

The `rcapd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/rcap:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

rctladm – システムのリソース制御の大域状態の表示または変更

形式

`rctladm [-lu] [-e action] [-d action] [name...]`

機能説明

`rctladm` コマンドを使えば、稼働中のシステム上に存在するアクティブなリソース制御を検査および変更できます。リソース制御のインスタンスは「*rctl*」と記述します。`rctl` については、`setrctl(2)` を参照してください。Solaris オペレーティングシステムの現行リリースでサポートされている `rctl` の一覧については、`resource_controls(5)` を参照してください。`rctl` 違反のロギングをシステム全体で有効化/無効化したり、アクティブな `rctl` (とその状態) を一覧表示したりすることができます。

オプションなしの `rctladm` コマンドは、`-l` オプション付きの `rctladm` と同等です。次の `-l` に対する説明を参照してください。

オプション

次のオプションがサポートされています。

`-d` *action*
`-e` *action*

指定された `rctl` 上で大域アクションを無効化 (`-d`) または有効化 (`-e`) します。`rctl` が指定されなかった場合、何のアクションも実行されず、エラーステータスが返されます。無効化オプションで特殊なトークン `all` を使用すれば、特定のリソース制御上のすべての大域アクションを無効にすることができます。

`syslog` アクションには、特定の重要度レベルを割り当てることでアクションレベルを設定できます。それには、`syslog=`*level* と指定します。ここで、*level* は、`syslog(3C)` で有効な重要度レベルとして記載されている文字列トークンのうちの 1 つです。重要度レベルの共通の `LOG_` 接頭辞は、省略できます。すべての `rctl` が `syslog` アクションをサポートしているとは限りません。`resource_controls(5)` を参照してください。

`-l`

`rctl` に関する情報を一覧表示します。名前、大域イベントアクションと大域ステータス、および大域フラグが表示されます。1 つ以上の名前オペランドが指定された場合、それらの名前に一致する `rctl` だけが表示されます。

`-u`

`/etc/rctladm.conf` の内容に基づいてリソース制御を構成します。名前オペランドはすべて無視されます。

オペランド

次のオペランドがサポートされています。

name

操作対象となる `rctl` の名前。複数の `rctl` 名を指定できます。名前が 1 つも指定されず、かつ一覧表示アクションが指定された場合には、すべての `rctl` が表示されます。有効化/無効化アクションを指定する場合、1 つ以上の `rctl` 名を指定する必要があります。

使用例

例1 特定の違反に対するシステムロギングの有効化

次のコマンドは、task.max-lwps に対するすべての違反のシステムロギングを有効化します。

```
# rctladm -e syslog task.max-lwps
#
```

例2 特定リソースの現在のステータスの検査

次のコマンドは、task.max-lwps リソースの現在のステータスを検査します。

```
$ rctladm -l task.max-lwps
task.max-lwps               syslog=DEBUG
$
```

終了ステータス

次の終了値が返されます。

0
: 正常終了。

1
: 致命的なエラーが発生しました。処理が失敗したそれぞれのリソース制御を知らせるメッセージが、標準エラーに書き込まれます。オペランドに指定されたその他のリソース制御の処理は、すべて成功しました。

2
: 無効なコマンド行オプションが指定されました。

ファイル

/etc/rctladm.conf
: rctladm が実行されるたびに、rctladm.conf の内容が現在の構成に基づいて更新されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

setrctl(2), getrctl(2), prctl(1), rctlblk_get_global_flags(3C), rctlblk_get_global_action(3C), attributes(5), resource_controls(5)

注意事項

デフォルトでは、rctl の違反は大域ログ作成では記録されません。

名前

rdate – set system date from a remote host

形式

`rdate` *hostname*

機能説明

`rdate` sets the local date and time from the *hostname* given as an argument. You must have the authorization `solaris.system.date` on the local system. Typically, `rdate` is used in a startup script.

`rdate` requests are responded to by the "time" service on the specified host. To enable the "time" service, use the following commands:

```
svcadm enable time:stream
svcadm enable time:dgram
```

使用法

The `rdate` command is IPv6–enabled. See `ip6`(7P).

属性

See `attributes`(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

関連項目

`inetd`(1M), `inetd.conf`(4), `attributes`(5), `ip6`(7P)

名前

reboot – オペレーティングシステムのリブート

形式

```
/usr/sbin/reboot [-dlnq] [-f | -p] [boot_arguments]
```

```
/usr/sbin/reboot [-f [-e environment] | -p] [-dlnq] [boot_arguments]
```

機能説明

reboot ユーティリティはカーネルをリブートします。カーネルは PROM モニターによってメモリーに読み込まれ、読み込まれたカーネルに制御が渡されます。

x86 システムでは、-f フラグを指定した場合、実行中のカーネルによって次のカーネルがメモリーに読み込まれ、続いて新しく読み込まれたカーネルに制御が渡されます。この reboot の形式は、前述の「形式」の 2 番目に示されています。

スーパーユーザーはいつでも reboot を実行できますが、通常は shutdown(1M) を使用して、これからサービスがシャットダウンされることを、すべてのユーザーに事前に警告する必要があります。詳細については、shutdown(1M) を参照してください。

reboot ユーティリティは sync(1M) 操作をディスクに実行して、その後マルチユーザーのリブートが開始されます。詳細については、init(1M) を参照してください。x86 システムでは、リブートの成功を確実にするために、reboot が、ブートアーカイブを必要に応じて変更することがあります。

reboot ユーティリティは通常、リブートの記録をシステムログデーモン syslogd(1M) に送信し、シャットダウンの記録をログインアカウンティングファイル /var/adm/wtmpx に保存します。これらの処理を抑制するには、-n または -q オプションを指定します。

通常、システムはパワーアップ時やクラッシュ後にリブートします。

オプション

次のオプションがサポートされています。

-d
: リブートする前に、システムクラッシュダンプを行います。システムクラッシュダンプの設定については、dumpadm(1M) を参照してください。

-e
: -f を指定すると、指定されたブート環境でリブートします。

  このオプションは現在、x86 システムでのみ使用できます。

-f
: x86 システムの場合

  高速リブート、ファームウェアのバイパス、およびブートローダー。実行中のカーネルによって、新しいカーネルがメモリーに読み込まれ、新しく読み込まれたカーネルに制御が渡されます。ディスクまたはカーネルの引数を指定する場合は、ほかの起動引数の前にこれらを指定する必要があります。

  SPARC システムの場合:

一部の POST テストをスキップすることによりリブートを高速化します。

サービス svc:/system/boot-config:default がデフォルトで有効です。action_authorization および value_authorization に solaris.system.shutdown が必要です。config/fastreboot_default プロパティーが true に設定されている場合、reboot は reboot -f として動作します。このプロパティーの値は、svccfg(1M) および svcadm(1M) を使用して変更でき、デフォルトのリブート動作を制御できます。

詳細は、「使用例」を参照してください。

-l
システムログデーモン syslogd(1M) へのメッセージ (誰が reboot を実行したかを示す) の送信を抑制します。

-n
sync(2) の呼び出しは行わず、syslogd(1M) または /var/adm/wtmpx にリブートの記録を保存しないようにします。カーネルはリブート前にファイルシステムとの同期を取ろうとします。ただし、-d オプションも指定した場合は例外です。-d と -n を一緒に指定した場合、カーネルはファイルシステムとの同期を取りません。

-p
PROM でリブートします。このフラグは、system/boot-config サービスの config/fastreboot_default プロパティー設定で指定されたデフォルトのリブート動作を変更せずに、ファームウェアを通じてシステムをリブートする場合に使用できます。

-p オプションと -f オプションを同時に指定することはできません。

-q
即時。実行中のプロセスを停止せずに、ただちにリブートします。

オペランド

次のオペランドがサポートされています。

boot_arguments
オプションの *boot_arguments* 文字列を指定すると、uadmin(2) 関数に引数を指定できます。これらの引数はリブート時にブートプログラムとカーネルに渡されます。引数の形式と一覧については、boot(1M) のマニュアルページと kernel(1M) のマニュアルページを参照してください。これらの引数を指定する場合、引数間のスペースは、シェルのために引用符で囲まれていない限り、単一のスペースに置換されます。*boot_arguments* がハイフン (-) で始まる場合、区切り文字列 (2 つのハイフン) を boot_arguments のハイフンの前に指定して、reboot 引数リストの終わりを示す必要があります (「使用例」の項を参照)。

使用例

例 1　-r と -v 引数を boot に渡す

次の例では、区切り文字列「--」 (2 つのハイフン) を使用して、reboot のオプションを boot(1M) の引数から分離する必要があります。

例 1　-r と -v 引数を boot に渡す　　(続き)

```
example# reboot -dl -- -rv
```

例 2　特定のディスクとカーネルを使用してリブートする

次の例は、特定のディスクとカーネルを使用してリブートします。

```
example# reboot disk1 kernel.test/unix
```

例 3　高速でリブートする

次の例では、-f オプションを使用して、高速リブートを実行します。

サービス svc:/system/boot-config:default が有効になっており、プロパティー config/fastreboot_default が true に設定されている場合、-f オプションを省略できます。

x86 システムでは、次のコマンドが、GRUB (grub(5) を参照) メニューファイル menu.lst でのデフォルトエントリでリブートします。

```
example# reboot -f
```

次のコマンドは、別の UFS ルートディスクでリブートします。

```
example# reboot -f -- '/dev/dsk/c1d0s0'
```

次のコマンドは、別の ZFS ルートプールでリブートします。

```
example# reboot -f -- 'rpool/ROOT/root2'
```

次のコマンドは、-k オプションを付けて、同じディスク上の mykernel でリブートします。

```
example# reboot -f -- '/platform/i86pc/mykernel/amd64/unix -k'
```

次のコマンドは、/mnt にマウントされた別のルートディスクから、mykernel でリブートします。

```
example# reboot -f -- '/mnt/platform/i86pc/mykernel/amd64/unix -k'
```

次のコマンドは、second_root という名前の別のブート環境上の /platform/i86pc/kernel/$ISADIR/unix でリブートします。

```
example# reboot -f -e second_root
```

次のコマンドは、-kv オプションを付けて、同じカーネルでリブートします。

```
example# reboot -f -- '-kv'
```

次のコマンドは、デフォルトの高速リブート動作を無効にします。

```
example# svccfg -s "system/boot-config:default" \
setprop config/fastreboot_default=false
example# svcadm refresh svc:/system/boot-config:default
```

例 3 高速でリブートする (続き)

次のコマンドは、デフォルトの高速リブート動作を再度有効にします。

```
example# svccfg -s "system/boot-config:default" \
setprop config/fastreboot_default=true
example# svcadm refresh svc:/system/boot-config:default
```

例 4 特定の GRUB メニューでリブートする

次のコマンドは、GRUB メニューのエントリ 2 でリブートします。

```
example# bootadm list-menu
  the location for the active GRUB menu is: /rpool/boot/grub/menu.lst
  default 0
  timeout 10
  0 zfsbe1
  1 zfsbe1 failsafe
  2 zfsbe2
  3 zfsbe2 Solaris xVM
  4 zfsbe2 failsafe
example# reboot 2
```

ファイル

`/var/adm/wtmpx`
ログインアカウンティングファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

mdb(1), boot(1M), dumpadm(1M), fsck(1M), halt(1M), init(1M), kernel(1M), shutdown(1M), svcadm(1M), svccfg(1M), sync(1M), syslogd(1M), sync(2), uadmin(2), reboot(3C), attributes(5), grub(5)

注意事項

reboot ユーティリティは /etc/rc*num*.d 内のスクリプトや inittab(4) 内のシャットダウンアクションを実行しません。システムサービスを完全にシャットダウンするためには、shutdown(1M) または init(1M) を使用して Solaris システムをリブートします。

名前

rem_drv – システムからのデバイスドライバの削除

形式

rem_drv [-b *basedir*] [-C] [-n] *device_driver*

機能説明

rem_drv コマンドは、システムにデバイスドライバ *device_driver* が有効でなくなったことを通知します。可能な場合、rem_drv はメモリーから *device_driver* を読み込み解除します。rem_drv はまた、システムのドライバ構成ファイルを更新します。

rem_drv は、実行されると、次回のシステムリブート時に再構成ブートを自動的に実行します (kernel(1M) を参照)。

オプション

次のオプションがサポートされています。

-b *basedir*
: ディスクレスクライアントのルートディレクトリへのパスを設定します。クライアントに対して rem_drv を実行するためにサーバー上で使用されます。ドライバを読み込み解除するには、クライアントマシンをリブートする必要があります。

  注 – -b オプションを使って、非大域ゾーンのルートファイルシステムを参照しないでください。この操作によって、大域ゾーンのファイルシステムを損傷したり、大域ゾーンのセキュリティを損ねたり、非大域ゾーンのファイルシステムを損傷する可能性があります。zones(5) を参照してください。

-C
: 削除されるドライバにバインドされた、ぶら下がるデバイス属性ノードを削除します。これにより、ノードに対して行われたデバイスの所有権やアクセス権のカスタマイズが、ドライバが再度追加される場合に保持されなくなります。ある構成からマシンを再プロビジョニングする場合に使用するか、これまでの管理上のカスタマイズが必要とされない構成に対して使用することをお勧めします。

-n
: *device_driver* の切り離しや読み込み解除を試みず、そのドライバのシステム構成ファイルの変更だけを行います。

使用例

例 1 sd ドライバの削除

次の例は、sd ドライバを削除して使用できないようにします。

```
example% rem_drv sd
```

例 2 ディスクレスクライアントの削除

次の例は、sun1 ディスクレスクライアントからドライバを削除します。ドライバは、このクライアントマシンがリブートされるまで、アンインストールまたは読み込み解除されません。

```
example% rem_drv -b /export/root/sun1 sd
```

-b オプションの使い方については、前述されているこのオプションへの注意事項に留意してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

add_drv(1M), kernel(1M), update_drv(1M), attributes(5), zones(5), devfs(7FS)

名前

remove_allocatable – 割り当てデータベースからのエントリの削除

形式

```
/usr/sbin/remove_allocatable  [-f] -n name

/usr/sbin/remove_allocatable  [-f] [-d] -t dev-type
```

機能説明

`remove_allocatable` は、デバイス割り当てメカニズムからユーザーが割り当て可能なデバイスのエントリを削除します。`remove_allocatable` は、このメカニズムによりラベル範囲が管理される、一部の割り当て不可能なデバイス (プリンタなど) のエントリも削除します。

オプション

次のオプションがサポートされています。

`-d` `-t` で指定されたデバイスタイプの、システム提供のデフォルト属性を削除します。

`-f` エントリを強制的に削除します。このオプションを指定しない場合、指定したデバイス名を持つエントリがもはや存在しないと、`remove_allocatable` はエラーを出力して終了します。

`-n` *name* デバイス *name* のエントリを削除します。

`-t` *dev-type* デバイスタイプが *dev-type* であるデバイスを削除します。

終了ステータス

成功した場合、`remove_allocatable` は終了ステータス `0` (真) を返します。エラーが発生した場合、`remove_allocatable` は、0 以外の終了ステータスを返します。終了コードとその意味を次に示します。

`1` 呼び出し時の構文エラー

`2` 原因不明のシステムエラー

`3` デバイス *name* または *dev-type* が見つからない。このエラーは、`-f` オプションが指定されていない場合にのみ発生します。

`4` アクセス権が与えられていません。ユーザーは、データベースに対する DAC または MAC アクセス権を持っていない。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

呼び出しは不確実です。オプションは不確実です。出力は非インタフェースです。

関連項目 allocate(1), deallocate(1), add_allocatable(1M), attributes(5), device_clean(5)

注意事項 このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

名前

removef – remove a file from software database

形式

```
removef [ [-M] -R root_path] [-V fs_file] pkginst path...
removef [ [-M] -R root_path] [-V fs_file] -f pkginst
```

機能説明

`removef` informs the system that the user, or software, intends to remove a pathname. Output from `removef` is the list of input pathnames that may be safely removed (no other packages have a dependency on them).

オプション

The following options are supported:

`-f`
: After all files have been processed, `removef` should be invoked with the `-f` option to indicate that the removal phase is complete.

`-M`
: Instruct `removef` not to use the $*root_path*`/etc/vfstab` file for determining the client's mount points. This option assumes the mount points are correct on the server and it behaves consistently with Solaris 2.5 and earlier releases.

`-R` *root_path*
: Define the full path name of a directory to use as the *root_path*. All files, including package system information files, are relocated to a directory tree starting in the specified *root_path*. The *root_path* may be specified when installing to a client from a server (for example, `/export/root/client1`).

  `removef` inherits the value of the `PKG_INSTALL_ROOT` environment variable. (See ENVIRONMENT VARIABLES, below.) If `PKG_INSTALL_ROOT` is set, such as when the `-R` option is used with `pkgadd(1M)` or `pkgrm(1M)`

  注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-V` *fs_file*
: Specify an alternative *fs_file* to map the client's file systems. For example, used in situations where the $*root_path*`/etc/vfstab` file is non-existent or unreliable.

オペランド

The following operands are supported:

path
: The pathname to be removed.

pkginst
: The package instance from which the pathname is being removed.

使用例

例 1 Using removef

The following example uses the `removef` command in an optional pre-install script:

```
echo "The following files are no longer part of this package
      and are being removed."
removef $PKGINST /myapp/file1 /myapp/file2 |
while read pathname
do
     echo "$pathname"
     rm -f $pathname
done
removef -f $PKGINST || exit 2
```

環境変数

`removef` inherits the value of the following environment variable. This variable is set when `pkgadd(1M)` or `pkgrm(1M)`

`PKG_INSTALL_ROOT`
: If present, defines the full path name of a directory to use as the system's `PKG_INSTALL_ROOT` path. All product and package information files are then looked for in the directory tree, starting with the specified `PKG_INSTALL_ROOT` path. If not present, the default system path of `/` is used.

終了ステータス

`0`
: Successful completion.

`>0`
: An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`pkginfo(1)`, `pkgmk(1)`, `pkgparam(1)`, `pkgproto(1)`, `pkgtrans(1)`, `installf(1M)`, `pkgadd(1M)`, `pkgask(1M)`, `pkgchk(1M)`, `pkgrm(1M)`, `attributes(5)`, `largefile(5)`

『Application Packaging Developer's Guide』

注意事項

Package commands are `largefile(5)`-aware. They handle files larger than 2 GB in the same way they handle smaller files. In their current implementations, `pkgadd(1M)`, `pkgtrans(1)` and other package commands can process a datastream of up to 4 GB.

名前

reparsed – 再解析ポイントデーモン

形式

`/usr/lib/reparse/reparsed`

機能説明

`reparsed` デーモンは、カーネルアップコールを処理して、「再解析ポイント」を解釈します。再解析ポイントは、Solaris SMB および NFS ファイルサーバーでの Microsoft DFS 照会および NFS 照会サポートの基礎要素です。このデーモンを実行できるのは、root ユーザーまたは同等の特権を持つユーザーだけです。デーモンは SMF の制御下にあります。

終了ステータス

`0`
デーモンが正常に起動されました。

`>0`
デーモンの起動に失敗しました。

エラー情報は、レベル `LOG_ERR` で `syslog` に報告されます。

ファイル

`/usr/lib/reparsed/*.so.1`
`reparsed` のサービス別プラグイン。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	コンソリデーション非公開

関連項目

`nfsref(1M)`, `svcadm(1M)`, `syslogd(1M)`, `libreparse(3LIB)`, `attributes(5)`

注意事項

`reparsed` デーモンは手動で停止、起動、または再起動しないでください。デーモンの状態を変更する必要がある場合は、次のコマンドを使用します。

```
# svcadm disable svc:/system/filesystem/reparse
# svcadm enable svc:/system/filesystem/reparse
# svcadm restart svc:/system/filesystem/reparse
```

詳細は、`svcadm(1M)` を参照してください。

名前

repquota – summarize quotas for a ufs file system

形式

```
repquota [-v] filesystem...
repquota -a [-v]
```

機能説明

repquota prints a summary of the disk usage and quotas for the specified ufs file systems. The current number of files and amount of space (in kilobytes) is printed for each user along with any quotas created with edquota(1M).

The *filesystem* must have the file quotas in its root directory.

Only the super-user may view quotas which are not their own.

オプション

The following options are supported:

- -a　Report on all mounted ufs file systems that have rq in the mntopts field of the /etc/vfstab file.
- -v　Report quotas for all users, even those who do not consume resources.

使用法

See largefile(5) for the description of the behavior of repquota when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

edquota(1M), quota(1M), quotacheck(1M), quotaon(1M), attributes(5), largefile(5), quotactl(7I)

名前

rmmount – removable media mounter for CD-ROM, Jaz drive, and others

形式

```
/usr/sbin/rmmount [-D]
```

機能説明

The `rmmount` utility is a removable media mounter that is executed by volume management whenever a removable medium, such as a CD-ROM is inserted. Removable media is managed by an application or a volume manager. `rmmount` can also be called by using `volrmmount(1)`.

Upon insertion of a medium and following invocation of the `volcheck(1)` command, `rmmount` determines what type of file system (if any) is on that medium. If a file system is present, `rmmount` mounts the file system in one of the locations listed below.

For a CD-ROM or a DVD-ROM:

`/cdrom/cdrom0`	symbolic link to mounted CD-ROM in local CD-ROM drive
`/cdrom/CD-ROM_name`	mounted named CD-ROM
`/cdrom/CD-ROM_name/partition`	mounted named CD-ROM with partitioned file system
`/cdrom/unnamed_cdrom`	mounted unnamed CD-ROM

For a Zip drive:

`/rmdisk/zip0`	symbolic link to mounted Zip medium in local Zip drive
`/rmdisk/`*Zip_name*	mounted named Zip medium
`/rmdisk/`*Zip_name*`/partition`	mounted named Zip medium with partitioned file system
`/rmdisk/unnamed_zip`	mounted unnamed Zip medium

For a Jaz drive:

`/rmdisk/jaz0`	symbolic link to mounted Jaz medium in local Jaz drive
`/rmdisk/`*Jaz_name*	mounted named Jaz medium
`/rmdisk/`*Jaz_name*`/partition`	mounted named Jaz medium with partitioned file system
`/rmdisk/unnamed_Jaz`	mounted unnamed Jaz medium

For a generic “rmdisk” drive:

`/rmdisk/rmdisk0`	symbolic link to mounted removable medium in local removable medium drive
`/rmdisk/`*rmdisk_name*	mounted named removable medium
`/rmdisk/`*rmdisk_name*`/partition`	mounted named removable medium with partitioned file system

`/rmdisk/unnamed_rmdisk` mounted unnamed removable medium

If the media is read-only (for example, a CD-ROM), the file system is mounted read-only.

If a file system is not identified, `rmmount` does not mount a file system. See the 『Oracle Solaris の管理: 一般的なタスク』 for more information on the location of CD-ROM and other media without file systems.

If a file system type has been determined, it is then checked to see that it is "clean." If the file system is "dirty," `fsck -p` (see `fsck(1M)`) is run in an attempt to clean it. If `fsck` fails, the file system is mounted read-only.

After the mount is complete, "actions" associated with the media type are executed. These actions allow for the notification to other programs that new media are available.

Actions are executed in the order in which they appear in the configuration file. The action function can return either `1` or `0`. If it returns `0`, no further actions will be executed. This allows the function to control which applications are executed.

In order to execute an action, `rmmount` performs a `dlopen(3C)` on the shared object and calls the action function defined within it. The definition of the interface to actions can be found in `/usr/include/rmmount.h`.

File systems mounted by `rmmount` are always mounted with the `nosuid` flag set, thereby disabling setuid programs and access to block or character devices in that file system. Upon ejection, `rmmount` unmounts mounted file systems and executes actions associated with the media type. If a file system is "busy" (that is, it contains the current working directory of a live process), the ejection will fail.

オプション

`-D` Turn on the debugging output from the `rmmount` `dprintf` calls.

ファイル

`/usr/lib/rmmount/*.so.1` shared objects used by `rmmount`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/storage/media-volume-manager

関連項目

`volcheck(1)`, `volrmmount(1)`, `fsck(1M)`, `dlopen(3C)`, `attributes(5)`

『Oracle Solaris の管理: 一般的なタスク』

名前

rmt – remote magtape protocol module

形式

`/usr/sbin/rmt`

機能説明

`rmt` is a program used by the remote dump and restore programs in manipulating a magnetic tape drive through an interprocess communication connection. `rmt` is normally started up with an `rexec(3SOCKET)` or `rcmd(3SOCKET)` call.

The `rmt` program accepts requests that are specific to the manipulation of magnetic tapes, performs the commands, then responds with a status indication. All responses are in ASCII and in one of two forms. Successful commands have responses of:

`A`*number*`\n` where *number* is an ASCII representation of a decimal number.

Unsuccessful commands are responded to with:

`E`*error-number*`\n`*error-message*`\n` where *error-number* is one of the possible error numbers described in `Intro(3)`, and *error-message* is the corresponding error string as printed from a call to `perror(3C)`.

The protocol consists of the following commands:

`S\n` Return the status of the open device, as obtained with a `MTIOCGET` `ioctl` call. If the operation was successful, an "ack" is sent with the size of the status buffer, then the status buffer is sent (in binary).

`C`*device*`\n` Close the currently open device. The *device* specified is ignored.

`I`*operation*`\n`*count*`\n` Perform a `MTIOCOP` `ioctl(2)` command using the specified parameters. The parameters are interpreted as the ASCII representations of the decimal values to place in the *mt_op* and *mt_count* fields of the structure used in the `ioctl` call. When the operation is successful the return value is the *count* parameter.

`L`*offset*`\nwhence\n` Perform an `lseek(2)` operation using the specified parameters. The response value is returned from the `lseek` call.

`O`*device*`\n`*mode*`\n` Open the specified *device* using the indicated *mode*. *device* is a full pathname, and *mode* is an ASCII representation of a decimal number suitable for passing to `open(9E)`. If a device is already open, it is closed before a new open is performed.

`R`*count*`\n` Read *count* bytes of data from the open device. `rmt` performs the requested `read(9E)` and responds with `A`*count-read*`\n` if the read was successful; otherwise an error in standard format is returned. If the read was successful, the data read is sent.

`W`*count*`\n` Write data onto the open device. `rmt` reads *count* bytes from the connection, aborting if a premature EOF is encountered. The

response value is returned from the write(9E) call.

Any other command causes rmt to exit.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

関連項目

ufsdump(1M), ufsrestore(1M), Intro(3), ioctl(2), lseek(2), perror(3C), rcmd(3SOCKET), rexec(3SOCKET), attributes(5), mtio(7I), open(9E), read(9E), write(9E)

診断

All responses are of the form described above.

使用上の留意点

Do not use this for a remote file access protocol.

名前

rmvolmgr – HAL 対応リムーバブルボリュームマネージャー

形式

```
/usr/sbin/rmvolmgr [-chns]
```

機能説明

rmvolmgr コマンドは、リムーバブルメディアとホットプラグ可能なストレージのマウントおよびマウント解除を自動的に行うことができるボリュームマネージャーです。デフォルトのマウントポイントは /media です。

rmvolmgr は、Solaris オペレーティングシステムに付属する多数のハードウェア抽象化レイヤー (HAL) 対応ツールの 1 つです。hald(1M) を参照してください。

rmvolmgr の複数のインスタンスを同時に実行できます。rmvolmgr のシステムインスタンスは、デフォルトでサービス管理機能 (SMF) サービスとして実行されます (smf(5) を参照)。障害管理リソース識別子 (FMRI) は次のとおりです。

```
svc:/system/filesystem/rmvolmgr
```

rmvolmgr を .xinitrc ファイルまたは同様のセッション起動スクリプトに追加すれば、ユーザー独自のインスタンスを実行できます。この場合、*system* rmvolmgr インスタンスは、起動スクリプトの所有者であるユーザーに属するボリュームを管理しません。たとえば、ワークステーションコンソール (/dev/console) にログオンしたユーザーで、rmvolmgr の自身のインスタンスを呼び出したユーザーは、CD-ROM ドライブや、ローカル USB または FireWire ポートに接続したデバイスなどのローカルで接続したデバイスを所有します。

/media の下にボリュームをマウントする以外に、rmvolmgr は、/cdrom および /rmdisk の下にレガシーシンボリックリンクも作成します。

rmvolmgr はまた、/tmp/.removable の下の通知ファイルを管理することにより、CDE リムーバブルメディアインタフェースとの下位互換性を確保しています。この機能は、-c オプションを使用して無効にできます。

-c および -n オプションも、SMF プロパティーとして指定できます。「使用例」を参照してください。

オプション

次のオプションがサポートされています。

- -c　CDE 互換性を無効にします。
- -h　ヘルプ情報を表示して終了します。
- -n　レガシーマウントポイントシンボリックリンクを作成しません。
- -s　システムインスタンスモードで呼び出します。

使用例

例 1　SMF プロパティーを使用してオプションを設定する

次の svccfg(1M) コマンドおよびサブコマンドは、SMF プロパティーを使用して、-c および -n オプションを設定します。

例 1　SMF プロパティーを使用してオプションを設定する　　(続き)

```
example# svccfg
svc:> select rmvolmgr
svc:/system/filesystem/rmvolmgr> listprop rmvolmgr/*
rmvolmgr/legacy_mountpoints     boolean  true
rmvolmgr/cde_compatible         boolean  true
svc:/system/filesystem/rmvolmgr> setprop \
  rmvolmgr/legacy_mountpoints=false
svc:/system/filesystem/rmvolmgr> setprop rmvolmgr/cde_compatible=false
svc:/system/filesystem/rmvolmgr> exit
example#
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/media-volume-manager
インタフェースの安定性	流動的

関連項目

hald(1M), svccfg(1M), attributes(5), hal(5), smf(5)

名前

rndc – name server control utility

形式

```
rndc [-V] [-b src-addr] [-c config-file] [-k key-file] [-s server]
     [-p port] [-y key_id] command
```

機能説明

The `rndc` utility controls the operation of a name server. It supersedes the `ndc` utility that was provided in previous BIND releases. If `rndc` is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

The `rndc` utility communicates with the name server over a TCP connection, sending commands authenticated with digital signatures. The only supported authentication algorithm in the current versions of `rndc` and `named(1M)` is HMAC-MD5, which uses a shared secret on each end of the connection. This algorithm provides TSIG-style authentication for the command request and the name server's response. All commands sent over the channel must be signed by a *key_id* known to the server.

The `rndc` utility reads a configuration file to determine how to contact the name server and decide what algorithm and key it should use.

オプション

The following options are supported:

`-b` *source-address*
: Use *source-address* as the source address for the connection to the server. Multiple instances are permitted to allow setting of both the IPv4 and IPv6 source addresses.

`-c` *config-file*
: Use *config-file* as the configuration file instead of the default `/etc/rndc.conf`.

`-k` *key-file*
: Use *key-file* as the key file instead of the default, `/etc/rndc.key`. The key in `/etc/rndc.key` is used to authenticate commands sent to the server if the *config-file* does not exist.

`-s` *server*
: The *server* argument is the name or address of the server that matches a server statement in the configuration file for `rndc`. If no server is supplied on the command line, the host named by the default-server clause in the options statement of the `rndc` configuration file is used.

`-p` *port*
: Send commands to TCP port *port* instead of BIND 9's default control channel port, 953.

`-V`
: Enable verbose logging.

`-y` *key_id*
: Use the key *key_id* from the configuration file. The *key_id* argument must be known by `named` with the same algorithm and secret string for control message validation to succeed. If no *key_id* is specified, `rndc` will first look for a key clause in the server statement of the

server being used, or if no server statement is present for that host, then the `default-key` clause of the options statement. The configuration file contains shared secrets that are used to send authenticated control commands to name servers. It should therefore not have general read or write access.

For the complete set of commands supported by `rndc`, see the BIND 9 Administrator Reference Manual or run `rndc` without arguments to see its help message.

Limitations

The `rndc` utility does not support all the commands of the BIND 8 `ndc` utility.

There is no way to provide the shared secret for a *key_id* without using the configuration file.

Several error messages tend toward the cryptic.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/dns/bind
Interface Stability	Volatile

関連項目

`named(1M)`, `rndc-confgen(1M)`, `named.conf(4)`, `rndc.conf(4)`, `attributes(5)`

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at `https://www.isc.org/software/bind/documentation`.

名前

rndc-confgen – rndc key generation tool

形式

```
rndc-confgen [-ah] [-b keysize] [-c keyfile] [-k keyname]
     [-p port] [-r randomfile] [-s address] [-t chrootdir]
     [-u user]
```

機能説明

The `rndc-confgen` utility generates configuration files for `rndc(1M)`. This utility can be used as a convenient alternative to writing by hand the `rndc.conf(4)` file and the corresponding `controls` and `key` statements in `named.conf`. It can also be run with the `-a` option to set up a `rndc.key` file and avoid altogether the need for a `rndc.conf` file and a `controls` statement.

オプション

The following options are supported:

`-a`

Perform automatic `rndc` configuration. This option creates a file `rndc.key` in `/etc` (or however *sysconfdir* was specified when BIND was built) that is read by both `rndc` and `named(1M)` on startup. The `rndc.key` file defines a default command channel and authentication key allowing `rndc` to communicate with `named` with no further configuration.

Running `rndc-confgen` with `-a` specified allows BIND 9 and `rndc` to be used as drop-in replacements for BIND 8 and `ndc`, with no changes to the existing BIND 8 `named.conf` file.

If a more elaborate configuration than that generated by `rndc-confgen -a` is required, for example if `rndc` is to be used remotely, you should run `rndc-confgen` without the `-a` option and set up `rndc.conf` and `named.conf` files, as directed.

`-b` *keysize*

Specify the size of the authentication key in bits. The *keysize* argument must be between 1 and 512 bits; the default is 128.

`-c` *keyfile*

Used with the `-a` option to specify an alternate location for `rndc.key`.

`-h`

Print a short summary of the options and arguments to `rndc-confgen`.

`-k` *keyname*

Specify the key name of the `rndc` authentication key. The *keyname* argument must be a valid domain name. The default is `rndc-key`.

`-p` *port*

Specify the command channel port where `named` listens for connections from `rndc`. The default is 953.

`-r` *randomfile*

Specify a source of random data for generating the authorization. By default, `/dev/random` is used. The *randomdev* argument specifies the name of a character device or file containing random data to be used instead of the default. The special value `keyboard` indicates that keyboard input should be used.

-s *address*
Specify the IP address where named listens for command channel connections from rndc. The default is the loopback address 127.0.0.1.

-t *chrootdir*
Used with the -a option to specify a directory where named will run after the root directory is changed with chroot(2). An additional copy of the rndc.key will be written relative to this directory so that it will be found by the named in the new directory.

-u *user*
Used with the -a option to set the owner of the rndc.key file generated. If -t is also specified only the file in the chroot area has its owner changed.

使用例

例 1 Create Automatic rndc Configuration

The following command creates an automatic rndc configuration, so that rndc can be used immediately.

```
# rndc-confgen -a
```

例 2 Print a Sample rndc.conf File

The following command prints a sample rndc.conf file with corresponding controls and key statements. These statements can subsequently be manually inserted in the file named.conf.

```
# rndc-confgen
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/dns/bind
Interface Stability	Volatile

関連項目

chroot(2), named(1M), rndc(1M), rndc.conf(4), attributes(5)

See the BIND 9 Administrator's Reference Manual. As of the date of publication of this man page, this document is available at https://www.isc.org/software/bind/documentation.

名前

roleadd – administer a new role account on the system

形式

```
roleadd [-c comment] [-d dir] [-e expire] [-f inactive]
      [-g group] [-G group [, group...]] [-m [-k skel_dir]]
      [-u uid [-o]] [-s shell] [-S repository]
      [-A authorization [,authorization...]] [-K key=value] role

roleadd -D [-b base_dir] [-e expire] [-f inactive]
      [-g group] [-A authorization [,authorization...]]
      [-P profile [,profile...] [-K key=value]]
```

機能説明

`roleadd` adds a role entry to the `passwd` and `shadow` and `user_attr` databases specified by the `-S` option. The default repository is `files`. The `-A` and `-P` options respectively assign authorizations and profiles to the role. Roles cannot be assigned to other roles. The `-K` option adds a *key=value* pair to `user_attr` for a role. Multiple *key=value* pairs can be added with multiple `-K` options.

`roleadd` also creates supplementary group memberships for the role (`-G` option) and creates the home directory (`-m` option) for the role if requested. The new role account remains locked until the `passwd(1)` command is executed.

Specifying `roleadd -D` with the `-g`, `-b`, `-f`, `-e`, or `-K` option (or any combination of these option) sets the default values for the respective fields. See the `-D` option. Subsequent `roleadd` commands without the `-D` option use these arguments.

The system file entries created with this command have a limit of 512 characters per line. Specifying long arguments to several options can exceed this limit.

The role (`role`) field accepts a string of no more than eight bytes consisting of characters from the set of alphabetic characters, numeric characters, period (`.`), underscore (_), and hyphen (`-`). The first character should be alphabetic and the field should contain at least one lower case alphabetic character. A warning message is written if these restrictions are not met. A future Solaris release might refuse to accept role fields that do not meet these requirements.

The `role` field must contain at least one character and must not contain a colon (`:`) or a newline (`\n`).

An administrator must be granted the User Management Profile to be able to create a new role. The authorizations required to set the various fields in `passwd`, `shadow` and `user_attr` can be found in `passwd(4)`, `shadow(4)`, and `user_attr(4)`. The authorizations required to assign groups can be found in `group(4)`.

オプション

The following options are supported:

`-A` *authorization*
: One or more comma separated authorizations defined in `auth_attr(4)`. Only a user or role who has grant rights to the authorization can assign it to an account

`-b` *base_dir*
: The default base directory for the system if `-d` *dir* is not specified. *base_dir* is concatenated with the account name to define the home directory. If the `-m` option is not used, *base_dir* must exist.

`-c` *comment*
: Any text string. It is generally a short description of the role. This information is stored in the role's `passwd` entry.

`-d` *dir*
: The home directory of the new role. It defaults to *base_dir*/*account_name*, where *base_dir* is the base directory for new login home directories and *account_name* is the new role name. This creates an `auto_home` entry for the user.

  The argument to the option can be specified as *server*:*dir* where *server* is the hostname of the machine on which the home directory resides and *dir* is the path to the user's home directory. If the server is a remote host then the home directory needs to be created on the remote host for the system to mount it, when the user logs in. If no server name is specified then the home directory will be created on the host where the command is executed, when the `-m` option is used.

`-D`
: Display the default values for *group*, *base_dir*, *skel_dir*, *shell*, *inactive*, *expire* and *key=value* pairs. When used with the `-g`, `-b`, `-f`, or `-K`, options, the `-D` option sets the default values for the specified fields. The default values are:

  group
  : `other` (GID of 1)

  base_dir
  : `/export/home`

  skel_dir
  : `/etc/skel`

  shell
  : `/bin/pfsh`

  inactive
  : `0`

  expire
  : Null

  auths
  : Null

  profiles
  : Null

key=value (pairs defined in user_attr(4)
not present

-e *expire*
Specify the expiration date for a role. After this date, no user is able to access this role. The expire option argument is a date entered using one of the date formats included in the template file /etc/datemsk. See getdate(3C).

If the date format that you choose includes spaces, it must be quoted. For example, you can enter 10/6/90 or October 6, 1990. A null value (" ") defeats the status of the expired date. This option is useful for creating temporary roles.

-f *inactive*
The maximum number of days allowed between uses of a role ID before that ID is declared invalid. Normal values are positive integers. A value of 0 defeats the status.

-g *group*
An existing group's integer ID or character-string name. Without the -D option, it defines the new role's primary group membership and defaults to the default group. You can reset this default value by invoking roleadd -D -g *group*.

-G *group*
An existing group's integer ID or character-string name. It defines the new role's supplementary group membership. Duplicates between *group* with the -g and -G options are ignored. No more than NGROUPS_MAX groups can be specified.

-k *skel_dir*
A directory that contains skeleton information (such as .profile) that can be copied into a new role's home directory. This directory must already exist. The system provides the /etc/skel directory that can be used for this purpose.

-K *key=value*
A *key=value* pair to add to the role's attributes. Multiple -K options can be used to add multiple *key=value* pairs. The generic -K option with the appropriate key can be used instead of the specific implied key options (-A and -P). See user_attr(4) for a list of valid *key=value* pairs. The “type” key is not a valid key for this option. Keys can not be repeated.

-m
Create the new role's home directory if it does not already exist. If the directory already exists, it must have read, write, and execute permissions by *group*, where *group* is the role's primary group. If the server name specified to the -d option is a remote host then the system will not attempt to create the home directory.

If the directory does not already exist, a new ZFS dataset will be created. In the global zone, the dataset is created as rpool/export/home/*rolename*. For non-global zones, the dataset will be created as *ROOT-dataset*/export/home/*rolename*. The mountpoint for the ZFS dataset is /export/home/*rolename* by default. If -d *path* is specified and it is a path on the local machine, the dataset will be mounted at the specified location. The role is delegated

permissions to create ZFS snapshots and promote them. The newly created dataset will inherit the encryption setting from its parent. If it is encrypted, the role is granted permission to change its wrapping key.

`-o`
This option allows a UID to be duplicated (non-unique).

`-P` *profile*
One or more comma-separated execution profiles defined in `prof_attr(4)`.

`-s` *shell*
Full pathname of the program used as the user's shell on login. It defaults to an empty field causing the system to use `/bin/pfsh` as the default. The value of *shell* must be a valid executable file.

`-S` *repository*
The valid repositories are `files`, `ldap`. The repository specifies which name service will be updated. The default repository is `files`. When the repository is `files`, the authorizations, profiles, and roles can be present in other name service repositories and can be assigned to a user in the `files` repository. When the repository is `ldap`, all the assignable attributes must be present in the `ldap` repository.

`-u` *uid*
The UID of the new role. This UID must be a non-negative decimal integer below `MAXUID` as defined in `<sys/param.h>`. The UID defaults to the next available (unique) number above the highest number currently assigned. For example, if UIDs 100, 105, and 200 are assigned, the next default UID number is 201. (UIDs from `0-99` are reserved for possible use in future applications.)

ファイル

`/etc/datemsk`

`/etc/passwd`

`/etc/shadow`

`/etc/group`

`/etc/skel`

`/usr/include/limits.h`

`/etc/user_attr`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

passwd(1), pfexec(1), profiles(1), roles(1), users(1B), groupadd(1M), groupdel(1M), groupmod(1M), grpck(1M), logins(1M), pwck(1M), userdel(1M), usermod(1M), getdate(3C), auth_attr(4), group(4), passwd(4), prof_attr(4), shadow(4), user_attr(4), attributes(5)

診断

In case of an error, roleadd prints an error message and exits with a non-zero status.

The following indicates that login specified is already in use:

```
UX: roleadd: ERROR: login is already in use. Choose another.
```

The following indicates that the *uid* specified with the -u option is not unique:

```
UX: roleadd: ERROR: uid uid is already in use. Choose another.
```

The following indicates that the *group* specified with the -g option is already in use:

```
UX: roleadd: ERROR: group group does not exist. Choose another.
```

The following indicates that the *uid* specified with the -u option is in the range of reserved UIDs (from 0-99):

```
UX: roleadd: WARNING: uid uid is reserved.
```

The following indicates that the *uid* specified with the -u option exceeds MAXUID as defined in <sys/param.h>:

```
UX: roleadd: ERROR: uid uid is too big. Choose another.
```

The following indicates that the /etc/passwd or /etc/shadow files do not exist:

```
UX: roleadd: ERROR: Cannot update system files - login cannot be created.
```

The following indicates that the user executing the command does not have sufficient authorization to perform the operation:

```
UX: roleadd: ERROR: Permission denied.
```

名前

roledel – delete a role's login from the system

形式

`roledel [-r] [-S` *repository*`] `*role*

機能説明

The `roledel` utility deletes a role account from the system and makes the appropriate account-related changes to the system file and file system. `roledel` also removes the role from each user's list of assumable roles.

An administrator must be granted the User Management Profile to be able to delete an existing role.

オプション

The following options are supported:

`-r`
: Remove the role's home directory from the system. This directory must exist. The files and directories under the home directory will no longer be accessible following successful execution of the command. The ZFS dataset that was created for the role's home directory will be removed.

  The `auto_home` entry for the role will be deleted.

`-S` *repository*
: The valid repositories are `files`, `ldap`. The repository specifies which name service will be updated. The default repository is `files`.

オペランド

The following operands are supported:

role An existing role name to be deleted.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`2` Invalid command syntax. A usage message for the `roledel` command is displayed.

`6` The account to be removed does not exist.

`8` The account to be removed is in use.

`10` Cannot update the `/etc/group` or `/etc/user_attr` file but the login is removed from the `/etc/passwd` file.

`12` Cannot remove or otherwise modify the home directory.

ファイル

`/etc/passwd`	system password file
`/etc/shadow`	system file containing roles' encrypted passwords and related information
`/etc/group`	system file containing group definitions
`/etc/user_attr`	system file containing additional role attributes

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

auths(1), passwd(1), profiles(1), roles(1), users(1B), groupadd(1M), groupdel(1M), groupmod(1M), logins(1M), roleadd(1M), rolemod(1M), useradd(1M), userdel(1M), usermod(1M), passwd(4), prof_attr(4), user_attr(4), attributes(5)

注意事項

The roledel utility only deletes an account definition that is in the group, passwd, shadow, and user_attr databases in the repository.

名前

rolemod – modify a role's login information on the system

形式

```
rolemod [-u uid [-o]] [-g group] [-G [+|-]group [, group...]]
     [-d dir [-m]] [-s shell] [-c comment] [-l new_name]
     [-f inactive] [-e expire]
     [-A [+|-]authorization [, authorization]]
     [-S repository]
     [-P [+|-]profile [, profile]] [-K key[+|-]=value] role
```

機能説明

The `rolemod` utility modifies a role's login information on the system. It changes the definition of the specified login and makes the appropriate login-related system file and file system changes.

The system file entries created with this command have a limit of 512 characters per line. Specifying long arguments to several options may exceed this limit.

An administrator must be granted the User Security Profile to be able to modify the security attributes for an existing role. To be able to modify non-security attributes of an existing user requires the User Management Profile. The authorizations required to set the various fields in `passwd`, `shadow` and `user_attr` can be found in `passwd(4)`, `shadow(4)`, `user_attr(4)`. The authorizations required to assign groups can be found in `group(4)`.

オプション

The following options are supported:

`-A` [+|]*authorization*

One or more comma separated authorizations as deined in `auth_attr(4)`. Only role with `grant` rights to the `authorization` can assign it to an account. This replaces any existing authorization setting. If no authorization list is specified, the existing setting is removed.

A prefix `+` adds the authorization to the existing authorization; a prefix `-` removes the authorization from the existing authorization. With no prefix, the value for *authorization* replaces the existing authorization.

`-c` *comment*

Specify a comment string. *comment* can be any text string. It is generally a short description of the login, and is currently used as the field for the user's full name. This information is stored in the user's `/etc/passwd` entry.

`-d` *dir*

Specify the new home directory of the role. It defaults to *base_dir/login*, where *base_dir* is the base directory for new login home directories, and `login` is the new login. This creates or modifies an `auto_home` entry for the user.

The argument to the option can be specified as *server*:*dir* where *server* is the hostname of the machine on which the home directory resides and *dir* is the path to the user's home directory. If the server is a remote host then the home directory needs to be created on the remote host for the system to mount it, when the user logs in. If no server name is specified then the home directory will be created on the host where the command is executed, when the `-m` option is used.

`-e` *expire*
Specify the expiration date for a role. After this date, no role will be able to access this login. The expire option argument is a date entered using one of the date formats included in the template file `/etc/datemsk`. See `getdate(3C)`.

For example, you may enter `10/6/90` or `October 6, 1990`. A value of “ `''` defeats the status of the expired date.

`-f` *inactive*
Specify the maximum number of days allowed between uses of a login ID before that login ID is declared invalid. Normal values are positive integers. A value of `0` defeats the status.

`-g` *group*
Specify an existing group's integer ID or character-string name. It redefines the role's primary group membership.

`-G` [+|-]*group*
An existing group's integer ID or character-string name. It defines the new user's supplementary group membership. Duplicates between group with the `-g` and `-G` options are ignored. No more than `NGROUPS_MAX` groups can be specified. GIDs 0-99 are reserved for allocation by the Solaris Operating System.

A prefix + adds the group to the existing group; a prefix - removes the group from the existing group. With no prefix, *group* replaces the existing group.

`-K` *key*[+|-]=*value*
Replace existing or add to a role's *key*=*value* pair attributes. Multiple `-K` options can be used to replace or add multiple *key*=*value* pairs. However, keys must not be repeated. The generic `-K` option with the appropriate key may be used instead of the specific implied key options (`-A` and `-P`). See `user_attr(4)` for a list of valid *key*=*value* pairs. If no value is specified, the existing key is removed.

The keyword `type` can be specified with the value `role` or the value `normal`. When using the value `normal`, the account changes from a role user to a normal user; using the value `role` keeps the account a role user.

A prefix + adds the value to the existing value; a prefix - removes the value from the existing value. With no prefix, *value* replaces the existing value.

The prefix +/- operation is applicable only to the following keys: `auths`, `profiles`, `roles`, `project`, `limitpriv`, and `defaultpriv`.

`-l` *new_logname*
Specify the new login name for the role. The *new_logname* argument is a string no more than eight bytes consisting of characters from the set of alphabetic characters, numeric characters, period (`.`), underline (_), and hypen (–). The first character should be alphabetic and the field should contain at least one lower case alphabetic character. A warning message will be written if these restrictions are not met. A future Solaris release

may refuse to accept login fields that do not meet these requirements. The *new_logname* argument must contain at least one character and must not contain a colon (`:`) or NEWLINE (`\n`).

`-m`
Move the role's home directory to the new directory specified with the `-d` option. If the directory already exists, it must have permissions read/write/execute by *group*, where *group* is the role's primary group. If the server name specified to the `-d` option is a remote host then the system will not attempt to create the home directory.

If the directory does not already exist, a new ZFS dataset will be created. In the global zone, the dataset is created as `rpool/export/home/`*rolename*. For non-global zones, the dataset will be created as *ROOT-dataset*`/export/home/`*rolename*. The mountpoint for the ZFS dataset is `/export/home/`*rolename* by default. If `-d` *path* is specified and it is a path on the local machine, the dataset will be mounted at the specified location. The role is delegated permissions to create ZFS snapshots and promote them. The newly created dataset will inherit the encryption setting from its parent. If it is encrypted, the role is granted permission to change its wrapping key.

`-o`
This option allows the specified UID to be duplicated (non-unique).

`-P` [+|-]*profile*
One or more comma-separated execution profiles defined in `auth_attr(4)`. This replaces any existing profile setting. If no profile list is specified, the existing setting is removed.

A prefix `+` adds the profile to the existing profile; a prefix `-` removes the profile from the existing profile. With no prefix, *profile* replaces the existing profile.

`-s` *shell*
Specify the full pathname of the program that is used as the role's shell on login. The value of *shell* must be a valid executable file.

`-S` *repository*
The valid repositories are `files`, `ldap`. The repository specifies which name service will be updated. The default repository is `files`. When the repository is `files`, the authorizations, profiles, and roles can be present in other name service repositories and can be assigned to a user in the `files` repository. When the repository is `ldap`, all the assignable attributes must be present in the `ldap` repository.

`-u` *uid*
Specify a new UID for the role. It must be a non-negative decimal integer less than `MAXUID` as defined in `<param.h>`. The UID associated with the role's home directory is not modified with this option; a role will not have access to their home directory until the UID is manually reassigned using `chown(1)`.

オペランド

The following operands are supported:

`login`
An existing login name to be modified.

終了ステータス

In case of an error, `rolemod` prints an error message and exits with one of the following values:

`2`
The command syntax was invalid. A usage message for the `rolemod` command is displayed.

`3`
An invalid argument was provided to an option.

`4`
The *uid* given with the `-u` option is already in use.

`5`
The password files contain an error. `pwconv(1M)` can be used to correct possible errors. See `passwd(4)`.

`6`
The login to be modified does not exist, the *group* does not exist, or the login shell does not exist.

`8`
The login to be modified is in use.

`9`
The *new_logname* is already in use.

`10`
Cannot update the `/etc/group` or `/etc/user_attr` file. Other update requests will be implemented.

`11`
Insufficient space to move the home directory (`-m` option). Other update requests will be implemented.

`12`
Unable to complete the move of the home directory to the new home directory.

ファイル

`/etc/group`
system file containing group definitions

`/etc/datemsk`
system file of date formats

`/etc/passwd`
system password file

`/etc/shadow`
system file containing users' and roles' encrypted passwords and related information

`/etc/user_attr`
: system file containing additional user and role attributes

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`chown(1)`, `passwd(1)`, `users(1B)`, `groupadd(1M)`, `groupdel(1M)`, `groupmod(1M)`, `logins(1M)`, `pwconv(1M)`, `roleadd(1M)`, `roledel(1M)`, `useradd(1M)`, `userdel(1M)`, `usermod(1M)`, `getdate(3C)`, `auth_attr(4)`, `group(4)`, `passwd(4)`, `shadow(4)`, `user_attr(4)`, `attributes(5)`

名前

root_archive – manage bootable miniroot archives

形式

`/boot/solaris/bin/root_archive pack` *archive root*

`/boot/solaris/bin/root_archive unpack` *archive root*

機能説明

The `root_archive` utility is used to manipulate boot archives. The utility can pack and unpack boot archives and image miniroots. Both `ufs` and `hsfs` (`iso9660`) format archives can be unpacked, although only `ufs` format is generated when packing.

For normal, boot-related system administration, `bootadm(1M)` is recommended.

Subcommands

The `root_archive` command has the following subcommands:

`pack` *archive root* Pack the contents of the root directory into the boot archive *archive*.

`unpack` *archive root* Unpack the contents of the boot archive named *archive* to the directory named *root*.

終了ステータス

The following exit values are returned:

`0` The command completed successfully.

`1` The command exited due to an error.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`cpio(1)`, `bootadm(1M)`, `mount(1M)`, `attributes(5)`, `lofi(7D)`

名前

route – ルーティングテーブルの手動操作

形式

```
route [-fnvq] sub-command [ [modifiers] args]

route [-fnvq] [-p [-R root-dir]] add | delete [modifiers] destination gateway
     [args]

route [-fnvq] change | get [modifiers] destination
     [gateway [args]]

route [-fn] monitor [modifiers]

route [-fnvq] flush [modifiers]

route -p [-R root-dir] show
```

機能説明

route は、ネットワークルーティングテーブルを手動で操作するためのユーティリティーです。これらのテーブルは通常、in.routed(1M) や in.ripngd(1M) などのシステムルーティングデーモンによって保守されます。

route でサポートされる一般的なオプションの数は限られていますが、強力なコマンド言語がサポートされます。ユーザーは、route(7P) で説明されているプログラム的なインタフェースによって実現できる任意の要求を指定できます。

route では、ルーティングソケットと、新しいメッセージタイプ RTM_ADD、RTM_DELETE、RTM_GET、および RTM_CHANGE を使用します。ルーティングテーブルを変更できるのはスーパーユーザーのみですが、RTM_GET 操作は非特権ユーザーにも許可されます。

オプション

-f
: すべてのゲートウェイエントリのルーティングテーブルをフラッシュします。-f オプションを任意の route サブコマンドと組み合わせて使用した場合、route は、それらのサブコマンドを実行する前にゲートウェイをフラッシュします。フラッシュするテーブルは、-f オプションの直後に inet または inet6 修飾子を置くことによって指定します。指定されていない場合は、デフォルトで IPv4 (inet) 経路がフラッシュされます。

-n
: アクションの報告時に、ホスト名やネットワーク名をシンボルで出力できないようにします。このオプションは、ネームサーバーが使用できない場合に有効です。

-p
: ネットワークルーティングテーブルへの変更が、システムの再起動後も永続するようにします。この操作は最初、ネットワークルーティングテーブルに適用され、成功すると、次にシステムの起動時に使用される保存された経路のリストに適用されます。 操作が成功したかどうかの判定では、すでに存在する経路を追加したり、ルーティングテーブルにない経路を削除したりしたときの失敗は無視されます。起動時に経路が追加される時点ではネットワークベースの名前解決サービスを使用できないため、永続的な経路にホスト名またはネットワーク名を使用する場合は特別な注意が必要です。

-q　すべての出力を抑制します。

-R *root-dir*　route が変更を適用する代替ルートディレクトリを指定します。このオプションは、-p オプションと組み合わせて使用されないかぎり無視されます。-R が指定されている場合、経路の変更は、起動時に使用される保存された経路のリストにのみ適用され、ネットワークルーティングテーブルには適用されません。さらに、-ifp で使用されるネットワークインタフェースの存在などの、特定のチェックもスキップされます。このオプションは、変更されるシステムのルートディレクトリが / 以外の場所にある JumpStart スクリプト内から使用する場合に便利です。

-v　詳細情報を冗長モードで出力します。

サブコマンド

サポートされているサブコマンドは次のとおりです。

add　経路を追加します。

change　経路の各側面 (ゲートウェイなど) を変更します。

delete　特定の経路を削除します。

flush　ルーティングテーブルからすべてのゲートウェイエントリを削除します。

get　宛先の経路を検索および表示します。

monitor　ルーティング情報ベースへの任意の変更、ルーティング検索の失敗、またはネットワークパーティション分割と考えられるものを継続的に報告します。

show　システムの起動時に適用される経路のリストを表示します。-p オプションとの組み合わせでのみ使用できます。

add および delete サブコマンドの構文は次のとおりです。

route [-fnvq] cmd *destination gateway* [metric/netmask]

ここで、*cmd* は add または delete、*destination* は宛先のホストまたはネットワーク、*gateway* はパケットが経路指定されるときに使用される次のホップの中継点です。「オペランド」で説明する修飾子は、コマンド行の任意の位置に指定できます。

get および change サブコマンドの構文は次のとおりです。

route [-fnvq] *cmd destination* [*gateway* [metric/netmask]]

ここで、*cmd* は get または change、*destination* は宛先のホストまたはネットワーク、*gateway* はパケットが経路指定されるときに使用される次のホップの中継点です。「オペランド」で説明する修飾子は、コマンド行の任意の位置に指定できます。

monitor サブコマンドの構文は次のとおりです。

```
route monitor [ -inet | -inet6 ]
```

オペランド

route は、これらのサブコマンドを、ゲートウェイを経由した宛先までの経路上で実行します。

宛先とゲートウェイ

デフォルトでは、宛先とゲートウェイアドレスは IPv4 アドレスとして解釈されます。すべてのシンボル名が最初、getipnodebyname(3SOCKET) を使用してホスト名として試行されます。AF_INET の場合にこの検索が失敗すると、getnetbyname(3SOCKET) は、この名前をネットワークの名前として解釈します。

コマンド行のアドレスの前にオプションの修飾子を指定すると、route サブコマンドでのアドレスの解釈方法が変更されます。

次の修飾子がサポートされています。

-inet　アドレスが強制的に IPv4 アドレスとして、つまり、AF_INET アドレスファミリの下で解釈されるようにします。

-inet6　アドレスが強制的に IPv6 アドレスとして、つまり、AF_INET6 アドレスファミリの下で解釈されるようにします。

IPv4 アドレスの場合は、宛先として指定されたインターネットアドレスを解釈することによって、特定のホストへの経路がネットワークへの経路とデフォルトで区別されます。宛先に 0 のローカルアドレス部分 (つまり、ネットマスクによってカバーされていない部分) が含まれている場合や、宛先がネットワークのシンボル名として解決された場合は、ネットワークへの経路であると見なされます。それ以外の場合は、ホストへの経路であると見なされます。

次のいずれかの修飾子を使用して、この選択を強制的に行うことができます。

-host　宛先が強制的にホストとして解釈されるようにします。

-net　宛先が強制的にネットワークとして解釈されるようにします。

例:

宛先	同等の宛先
128.32	-host 128.0.0.32

宛先	同等の宛先
128.32.130	-host 128.32.0.130
-net 128.32	128.32.0.0
-net 128.32.130	128.32.130.0

2 つの修飾子によって、アドレスとキーワードの間の混乱 (たとえば、host がホストのシンボル名として使用されている場合) が回避されます。*destination* の前に -dst 修飾子を指定することによって、宛先を識別できます。-gateway 修飾子を使用してゲートウェイアドレスを識別できます。ゲートウェイとして機能する中間の IP ルーターを必要としないインタフェースを経由して宛先に直接到達可能な場合は、これを -interface または -iface 修飾子を使用して示すことができます。

次の例では、経路は外部のゲートウェイ (ルーター) ではなく、マシンのいずれかのインタフェースを指しています。このような経路上の宛先とマスクに一致する IP 宛先アドレスを含むパケットが、ゲートウェイアドレスによって識別されたインタフェースから送信されます。 ARP プロトコルを使用するインタフェースの場合は、このタイプの経路を使用して、一致するすべての宛先が物理リンクに対してローカルであることを指定します。つまり、このコマンドを使用してデフォルトの経路を追加することによって、設定されているインタフェースネットマスクには関係なく、すべてのアドレスに対して ARP を使用するようにホストを設定できます。例:

```
example# route add default hostname -interface
```

ここで、ゲートウェイアドレス *hostname* は、一致するすべてのパケットの送信に使用されるネットワークインタフェースに関連付けられた名前または IP アドレスです。1 つのネットワークインタフェースを備えたホストでは、*hostname* は通常、uname -n コマンドによって返される *nodename* と同じです。uname(1) を参照してください。

古いシステムとの下位互換性のために、直接到達可能な経路は、ゲートウェイアドレスのあとに 0 を置くことによっても指定できます。

```
example# route add default hostname 0
```

この値は以前はルートメトリックでしたが、このメトリックは使用されなくなりました。この値が 0 として指定されている場合、宛先は直接到達可能です (-interface の指定と同等)。 この値が 0 ではないが、サブネットマスクとして解釈できない場合は、ゲートウェイが使用されます (デフォルト)。

AF_INET アドレスファミリまたは IPv4 アドレスでは、個別のサブネットマスクを指定できます。これは、次のいずれかの方法で指定できます。

- ゲートウェイアドレスのあとの IP アドレス。これは通常、シンボル形式ではなく、inet_addr(3SOCKET) の場合のような *10* 進数のドット表記法で指定されます。

- -netmask 修飾子のあとの IP アドレス。
- 宛先アドレスに追加されたスラッシュ文字と 10 進数の長さ。

サブネットマスクが指定されていないとき、宛先のクラスフルネットワークがインタフェースのクラスフルネットワークと同じである場合、使用されるマスクはゲートウェイアドレスによって選択された出力インタフェースのサブネットマスクです。それ以外の場合は、宛先アドレスのクラスフルネットワークマスクが使用されます。

次の各例は、255.255.255.224 のサブネットマスクを持つ宛先 192.0.2.32 サブネットへの IPv4 経路を作成します。

```
example# route add 192.0.2.32/27 somegateway
example# route add 192.0.2.32 -netmask 255.255.255.224 somegateway
example# route add 192.0.2.32 somegateway 255.255.255.224
```

IPv6 の場合は、スラッシュ形式のみが使用できます。次の例は、16 個のビット 1 のあとに 112 個のビット 0 が続くネットマスクを持つ宛先 33fe:: への IPv6 経路を作成します。

```
example# route add -inet6 3ffe::/16 somegateway
```

ゲートウェイによって出力インタフェースが一意に識別されない場合 (たとえば、複数のインタフェースが同じアドレスを持っている場合) は、-ifp *ifname* 修飾子を使用してインタフェースを名前で指定することができます。たとえば、-ifp lo0 は経路を lo0 インタフェースに関連付けます。指定されたインタフェースが IPMP (IP マルチパス) グループの基本となるインタフェースの場合、経路を追加する要求は対応する IPMP IP インタフェースに自動的に変換され、基本となるインタフェースの経路を削除または変更する要求は失敗します。

ルーティングテーブルに複数の同じ経路 (つまり同じ着信先とマスクの経路) が含まれている場合、IP はそれらの経路にトラフィックを分配することを試みます。分配では、個々の転送接続が同じ経路を使用して、たとえば TCP で見られるようなパケットの並び替えを防止します。分配のアルゴリズムの詳細はドキュメント化されておらず、随時変更される可能性があります。

ルーティングフラグ

経路には、各経路に一致する宛先に送信するときのプロトコルの動作に影響を与えるフラグが関連付けられています。これらのフラグは、コマンド行で次の修飾子を指定することによって設定 (場合によっては、~ で示すことによって消去) できます。

修飾子	フラグ	機能説明
-interface	~RTF_GATEWAY	宛先は直接到達可能です。
-iface	~RTF_GATEWAY	インタフェース修飾子の別名です。
-static	RTF_STATIC	手動で追加された経路です。

修飾子	フラグ	機能説明
-nostatic	~RTF_STATIC	経路がカーネルまたはルーティングデーモンによって追加されたように見せかけます。
-reject	RTF_REJECT	一致した場合に、ICMP 到達不能を発行します。
-blackhole	RTF_BLACKHOLE	パケットを暗黙的に破棄します。
-proto1	RTF_PROTO1	プロトコル固有のルーティングフラグ #1 を設定します。
-proto2	RTF_PROTO2	プロトコル固有のルーティングフラグ #2 を設定します。
-private	RTF_PRIVATE	この経路を通知しません。
-multirt	RTF_MULTIRT	指定された冗長経路を作成します。
-setsrc	RTF_SETSRC	デフォルトの発信元アドレスを割り当てます。
-indirect	RTF_INDIRECT	ゲートウェイがオンリンクではない場所への経路の追加を許可します。

任意指定の -indirect 修飾子は、ゲートウェイに直接到達できない場所への経路の追加を許可します。間接的な経路がパケットの送信または転送に最適である場合、IP はそのゲートウェイを検索して、直接到達可能な経路を探します。-indirect 修飾子は、ゲートウェイに直接到達可能な場合でも使用できます。

オプションの修飾子

-rtt、-rttvar、-sendpipe、-recvpipe、-mtu、-hopcount、-expire、および -ssthresh は、TCP などのトランスポートレベルのプロトコルによってルーティングエントリ内に保持される数量に初期値を提供します。これらの修飾子を個別にロックするには、ロックする各修飾子の前に -lock メタ修飾子を指定するか、またはあとに続くすべてのメトリックスを -lockrest メタ修飾子でロックすることを指定します。

一部のトランスポートレイヤープロトコルは、これらのメトリックスの一部のみをサポートできます。次のオプションの修飾子がサポートされています。

-expire　エントリの寿命。このオプションの修飾子は、現在サポートされていません。

-hopcount　最大ホップ数。このオプションの修飾子は、現在サポートされていません。

-mtu　最大 MTU (バイト単位)。

-recvpipe　受信パイプのサイズ (バイト単位)。

-rtt　往復時間 (マイクロ秒単位)。

-rttvar　往復時間の分散 (マイクロ秒単位)。

-sendpipe　送信パイプのサイズ (バイト単位)。

-ssthresh　送信パイプのサイズのしきい値 (バイト単位)。

-secattr　経路のセキュリティー属性。この修飾子を使用できるのは、システムに Solaris Trusted Extensions 機能が設定されている場合だけです。

-secattr 修飾子の形式は次のとおりです。

min_sl=*val*,max_sl=*val*,doi=*val*,cipso

または

sl=VAL,doi=VAL,cipso

上の最初の形式では、min_sl と max_sl の *val* は、16 進数または文字列のどちらかの形式の機密ラベルです。doi の *val* は、負でない整数です。経路は、doi 値で定義されているものと同じ解釈のドメインを持ち、かつ min_sl と max_sl の値で定義されている認可範囲内にあるパケットに対してのみ適用されます。cipso キーワードは省略可能で、デフォルトで設定されます。有効な min_sl、max_sl、および doi のキーワード/値のペアは必須です。*val* にスペースが含まれている場合は、二重引用符で保護されている必要があることに注意してください。

上の 2 番目の形式は、min_sl と max_sl に同じ VAL を使用して最初の形式を指定した場合と同等です。get コマンドでは 1 つの機密ラベルだけが使用されるため、2 番目の形式は get コマンドで使用してください。

互換性

修飾子 host および net は、-host および -net と同等と見なされます。これらのいずれかの名前に一致する記号アドレスを指定するには、dst または gateway キーワードを使用して区別します。たとえば、-dst host と指定します。

次の 2 つのフラグも、古いシステムとの互換性のために受け入れられますが、何も実行されません。

修飾子	フラグ
-cloning	RTF_CLONING
-xresolve	RTF_XRESOLVE

-ifa hostname 修飾子も受け入れられますが、何も実行されません。

ファイル

/etc/defaultrouter	デフォルト経路のリスト
/etc/hosts	ホスト名とネットアドレスのリスト
/etc/networks	ネットワークの名前とアドレスのリスト

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

uname(1), in.ripngd(1M), in.routed(1M), netstat(1M), routed(1M), ioctl(2), getipnodebyname(3SOCKET), getnetbyname(3SOCKET), inet_addr(3SOCKET), defaultrouter(4), hosts(4), networks(4), attributes(5), arp(7P), ip(7P), route(7P), routing(7P)

診断

add [host | network] *destination*:*gateway flags*

指定された経路がテーブルに追加されます。出力される値は、ioctl(2) の呼び出しで指定されたルーティングテーブルエントリの情報です。使用されたゲートウェイアドレスがゲートウェイのプライマリアドレス (getipnodebyname(3SOCKET) によって返される最初のアドレス) でなかった場合、ゲートウェイアドレスは数値とシンボルで出力されます。

delete [host | network] *destination*:*gateway flags*
change [host | network] *destination*:*gateway flags*

add と同様ですが、エントリを削除または変更する場合に使用されます。

destination done

-f フラグが指定されているか、または flush サブコマンドが使用されている場合は、削除された各ルーティングテーブルエントリがこの形式のメッセージで示されます。

Network is unreachable

指定されたゲートウェイが直接接続されたネットワーク上に存在しなかったため、経路を追加する試

	みが失敗しました。代わりに、次のホップのゲートウェイを指定します。
`not in table`	テーブルに存在しないエントリに対して `delete` 操作が試行されました。
`entry exists`	カーネル内にすでに存在する経路に対して `add` 操作が試行されました。
`routing table overflow`	操作が試行されましたが、システムは、新しいエントリを作成するためのメモリーを割り当てることができませんでした。
`insufficient privileges`	呼び出し側プロセスが適切な特権を持っていないため、経路を追加、削除、変更、またはフラッシュする試みが失敗しました。

注意事項

宛先がローカルであることを (`-interface` 修飾子で) 指定する場合は、ルーターに `proxy ARP` が実装されていることを前提にしています。つまり、ルーターは、すべての到達可能な宛先への ARP クエリーに応答します。通常は、proxy ARP を使用するよりも、ルーター検索または RIP のどちらかを使用する方が信頼性と拡張性に優れています。RIP に関連する情報については、`in.routed(1M)` を参照してください。

すべての宛先がローカルである経路とサブネットまたはネットワーク経路を組み合わせると、予期しない結果になることがあります。すべての宛先がローカルである経路に関連して行われる検索の順序は未定義であり、リリースによって変更されることがあります。

名前

routeadm – IP forwarding and routing configuration

形式

```
routeadm [-p [option]]

routeadm [-R root-dir] [-e option ...] [-d option...]
 [-r option...] [-s var=value]

routeadm [-l fmri]

routeadm [-m fmri key=value [key=value]...]

routeadm [-u]
```

機能説明

The `routeadm` command is used to administer system-wide configuration for IP forwarding and routing. IP forwarding is the passing of IP packets from one network to another; IP routing is the use of a routing protocol to determine routes.

IP routing functions are also represented as services within the service management facility (SMF), and can be administered by means of `svcadm(1M)` also, using the following fault management resource identifiers (FMRIs):

```
svc:/network/routing/route:default
svc:/network/routing/ripng:default
```

See EXAMPLES for relevant examples.

In addition to enabling and disabling routing and forwarding, `routeadm` is used to interact with SMF-based routing daemon services. Routing daemon services are identified by the presence of a `routeadm` application property group, which `routeadm` uses in administering the given service. Routing daemon services can also specify properties relating to their operation in the `routing` application property group; these can be modified by means of `routeadm -m`. If an FMRI for a service without such a property group is specified, an error is issued and the operation is not carried out. If a routing daemon has not been converted to SMF, the `ipv4`*[or 6]*`-routing-daemon`, `ipv4`*[or 6]*`-routing-daemon-args`, and `ipv4`*[or 6]*`-routing-stop-cmd` variables can be used to specify the appropriate daemon for IPv4 or IPv6 routing. `routeadm` will then run that daemon using the `svc:/network/routing/legacy-routing:ipv4`*[or 6]* service as appropriate. This conversion process occurs when you issue an enable (`-e`), disable (`-d`) or an update (`-u`) command.

The first usage, in the SYNOPSIS above, reports the current configuration.

オプション

The following command-line options are supported:

`-p` [*option*]
: Print the configuration in parseable format. If *option* is specified, only the configuration for the specified option or variable is displayed.

`-R` *root-dir*
Specify an alternate root directory where `routeadm` applies changes. This can be useful from within JumpStart scripts, where the root directory of the system being modified is mounted elsewhere.

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might compromise the security of the global zone, and might damage the non-global zone's file system. See `zones(5)`.

`-e` *option...*
Enable the specified option. The effect is to prepare the associated services for enabling. By means of the `routing-svcs` variable, the routing daemons are specified to be enabled on subsequent boot or when `routeadm -u` is run.

`-d` *option...*
Disable the specified option. The effect is to prepare the associated services for disabling. By means of the `routing-svcs` variable, the routing daemons are specified to be disabled on subsequent boot or when `routeadm -u` is run.

`-l` *fmri*
List all properties in the routing application property group for the SMF routing daemon service.

`-m` *fmri key=value*
Change property value of property *key* to *value* in routing application property group for the SMF routing daemon service. For multi-valued properties, the property name can be used multiple times in the modify operation, and each associated value will be added.

`-r` *option...*
Revert the specified option to the system default. The system defaults are specified in the description of each *option*.

`-u`
Apply the currently configured options to the running system. These options might include enabling or disabling IP forwarding and launching or killing routing daemons, if any are specified. It does not alter the state of the system for those settings that have been set to default. This option is meant to be used by administrators who do not want to reboot to apply their changes. In addition, this option upgrades non-SMF configurations from the invocations of daemon stop commands, which might include a set of arguments, to a simple enabling of the appropriate service.

`-s` *key=value*
Specify string values for specific variables in a comma-separated list with no intervening spaces. If invalid options are specified, a warning message is displayed and the program exits. The following variables can be specified:

`routing-svcs=`*fmrilist*
Specifies the routing daemon services to be enabled. Routing daemon services are determined to be IPv4 or IPv6 (and so enabled or disabled when `routeadm -e/-d`

ipv4*(6)*-routing is run) on the basis of property values in the routeadm application property group. Default: route:default ripng:default

ipv4-routing-daemon=<*full_path_to_routing_daemon*>
Specifies the routing daemon to be started when ipv4-routing is enabled. The routing daemon specified must be an executable binary or shell-script. If the specified program maps to an SMF service, the service will be used, and daemon arguments to the program will be transferred to the properties of the service at enable time. Default: ""

ipv4-routing-daemon-args=<*args*>
Specifies the startup arguments to be passed to the ipv4-routing-daemon when ipv4-routing is enabled. Default: no arguments

ipv4-routing-stop-cmd=<*command*>
Specifies the command to be executed to stop the routing daemon when ipv4-routing is disabled. <*command*> can be an executable binary or shell-script, or a string that can be parsed by system(3C). Default: ""

ipv6-routing-daemon=<*full_path_to_routing_daemon*>
Specifies the routing daemon to be started when ipv6-routing is enabled. The routing daemon specified must be an executable binary or shell-script. If the specified program maps to an SMF service, the service will be used, and daemon arguments to the program will be transferred to the properties of the service at enable time. Default: ""

ipv6-routing-daemon-args=<*args*>
Specifies the startup arguments to be passed to the ipv6-routing-daemon when ipv6-routing is enabled. Default: ""

ipv6-routing-stop-cmd=<*command*>
Specifies the command to be executed to stop the routing daemon when ipv6-routing is disabled. <*command*> can be an executable binary or shell-script, or a string that can be parsed by system(3C). Default: ""

Multiple -e, -d, and -r options can be specified on the command line. Changes made by -e, -d, and -r are persistent, but are not applied to the running system unless routeadm is called later with the -u option.

Use the following options as arguments to the -e, -d, and -r options (shown above as *option...*).

ipv4-forwarding
Controls the global forwarding configuration for all IPv4 interfaces. The system default is disabled. If enabled, IP will forward IPv4 packets to and from interfaces when appropriate. If disabled, IP will not forward IPv4 packets to and from interfaces when appropriate.

ipv4-routing
Determines whether an IPv4 routing daemon is run. The system default is enabled unless the /etc/defaultrouter file exists (see defaultrouter(4)), in which case the default is

disabled. The value of this option reflects the state of all IPv4 routing services, such that if any IPv4 routing service is enabled, ipv4-routing is enabled. This allows users to interact with routing services using svcadm(1M), as well as through routeadm. IPv4 routing services, specified by means of the routing-svcs variable, will be prepared for enable on next boot when the user explicitly enables ipv4-routing. The SMF routing daemon service for in.routed (svc:/network/routing/route:default) is specified by default.

ipv6-forwarding
: Controls the global forwarding configuration for all IPv6 interfaces. The system default is disabled. If enabled, IP will forward IPv6 packets to and from interfaces when appropriate. If disabled, IP will not forward IPv6 packets to and from interfaces when appropriate.

ipv6-routing
: Determines whether an IPv6 routing daemon is run. The system default is disabled. The value of this option reflects the state of all IPv6 routing services, such that, if any IPv6 routing service is enabled, ipv6-routing is enabled. This allows users to interact with routing services via svcadm(1M) as well as through routeadm. IPv6 routing services, specified by means of the routing-svcs variable, will be prepared for enable on next boot when the user explicitly enables ipv6-routing. The SMF routing daemon service for in.ripngd (svc:/network/routing/ripng:default) is specified by default.

The forwarding and routing settings are related but not mutually dependent. For example, a router typically forwards IP packets and uses a routing protocol, but nothing would prevent an administrator from configuring a router that forwards packets and does not use a routing protocol. In that case, the administrator would enable forwarding, disable routing, and populate the router's routing table with static routes.

The forwarding settings are global settings. Each interface also has an IFF_ROUTER forwarding flag that determines whether packets can be forwarded to or from a particular interface. That flag can be independently controlled by means of ifconfig(1M)'s router option. When the global forwarding setting is changed (that is, -u is issued to change the value from enabled to disabled or vice-versa), all interface flags in the system are changed simultaneously to reflect the new global policy. Interfaces configured by means of DHCP automatically have their interface-specific IFF_ROUTER flag cleared.

When a new interface is plumbed by means of ifconfig, the value of the interface-specific forwarding flag is set according to the current global forwarding value. Thus, the forwarding value forms the “default” for all new interfaces.

使用例

例 1 Enabling IPv4 Forwarding

IPv4 forwarding is disabled by default. The following command enables IPv4 forwarding:

```
example# routeadm -e ipv4-forwarding
```

例 2 Apply Configured Settings to the Running System

In the previous example, a system setting was changed, but will not take effect until the next reboot unless a command such as the following is used:

```
example# routeadm -u
```

例 3 Making a Setting Revert to its Default

To make the setting changed in the first example revert to its default, enter the following:

```
example# routeadm -r ipv4-forwarding
example# routeadm -u
```

例 4 Starting in.routed with the -q Flag

Setting the `-q` flag is represented in the SMF service by setting the `quiet_mode` property to true. The following sequence of commands starts `in.routed` with the `-q` flag:

```
example# routeadm -m route:default quiet_mode=true
example# routeadm -e ipv4-routing -u
```

See `in.routed(1M)` for details of property names and how they relate to daemon behavior.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`!=0` An error occurred while obtaining or modifying the system configuration.

ファイル

`/etc/inet/routing.conf` Parameters for IP forwarding and routing. (Not to be edited.)

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`ifconfig(1M)`, `in.routed(1M)`, `ipadm(1M)`, `svcadm(1M)`, `gateways(4)`, `attributes(5)`, `smf(5)`

名前

rpcbind – universal addresses to RPC program number mapper

形式

```
rpcbind [-d] [-w]
```

機能説明

rpcbind is a server that converts RPC program numbers into universal addresses. It must be running on the host to be able to make RPC calls on a server on that machine.

When an RPC service is started, it tells rpcbind the address at which it is listening, and the RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it first contacts rpcbind on the server machine to determine the address where RPC requests should be sent.

rpcbind should be started before any other RPC service. Normally, standard RPC servers are started by port monitors, so rpcbind must be started before port monitors are invoked.

When rpcbind is started, it checks that certain name-to-address translation-calls function correctly. If they fail, the network configuration databases can be corrupt. Since RPC services cannot function correctly in this situation, rpcbind reports the condition and terminates.

rpcbind maintains an open transport end for each transport that it uses for indirect calls. This is the UDP port on most systems.

The rpcbind service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rpc/bind
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). rpcbind can only be started by the superuser or someone in the Primary Administrator role.

The configuration properties of this service can be modified with svccfg(1M).

The following SMF property is used to allow or disallow access to rpcbind by remote clients:

```
config/local_only = true
```

The default value, true, shown above, disallows remote access; a value of false allows remove access. See EXAMPLES.

The FMRI svc:network/rpc/bind property group config contains the following property settings:

enable_tcpwrappers
: Specifies that the TCP wrappers facility is used to control access to TCP services. The value true enables checking. The default value for enable_tcpwrappers is false. If the enable_tcpwrappers parameter is enabled, then all calls to rpcbind originating from non-local addresses are automatically wrapped by the TCP wrappers facility. The syslog facility code daemon is used to log

allowed connections (using the `info` severity level) and denied traffic (using the `warning` severity level). See `syslog.conf(4)` for a description of `syslog` codes and severity levels. The Interface Stability of the TCP wrappers facility and its configuration files is Volatile. As the TCP wrappers facility is not controlled by Sun, intrarelease incompatibilities are not uncommon. See `attributes(5)`.

`verbose_logging` Specifies whether the TCP wrappers facility logs all calls orjust the denied calls. The default is `false`. This option has no effect if TCP wrappers are not enabled.

`allow_indirect` Specifies whether `rpcbind` allows indirect calls at all. By default, `rpcbind` allows most indirect calls, except to a number of standard services(`keyserv`, `automount`, `mount`, `nfs`, `rquota`, and selected NIS and `rpcbind` procedures). Setting `allow_indirect` to `false` causes all indirect calls to be dropped. The default is `true`. NIS broadcast clients rely on this functionality on NIS servers.

オプション

The following options are supported:

`-d` Run in debug mode. In this mode, `rpcbind` does not fork when it starts. It prints additional information during operation, and aborts on certain errors. With this option, the name-to-address translation consistency checks are shown in detail.

`-w` Do a warm start. If `rpcbind` aborts or terminates on `SIGINT` or `SIGTERM`, it writes the current list of registered services to `/var/run/portmap.file` and `/var/run/rpcbind.file`. Starting `rpcbind` with the `-w` option instructs it to look for these files and start operation with the registrations found in them. This allows `rpcbind` to resume operation without requiring all RPC services to be restarted.

使用例

例 1 Allowing Remote Access

The following sequence of commands allows remote access to `rpcbind`.

```
# svccfg -s svc:/network/rpc/bind setprop config/local_only = false
# svcadm refresh svc:/network/rpc/bind
```

ファイル

`/var/run/portmap.file` Stores the information for RPC services registered over IP based transports for warm start purposes.

`/var/run/rpcbind.file` Stores the information for all registered RPC services for warm start purposes.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

TCP wrappers is Volatile.

関連項目

smf(5), rpcinfo(1M), svcadm(1M), svccfg(1M), rpcbind(3NSL), syslog.conf(4), attributes(5), smf(5)

For information on the TCP wrappers facility, see the hosts_access(4) man page, delivered as part of the Solaris operating environment in /usr/sfw/man and available in the security/tcp-wrapper package.

注意事項

Terminating rpcbind with SIGKILL prevents the warm-start files from being written.

All RPC servers are restarted if the following occurs: rpcbind crashes (or is killed with SIGKILL) and is unable to to write the warm-start files; rpcbind is started without the -w option after a graceful termination. Otherwise, the warm start files are not found by rpcbind.

名前

rpc.bootparamd, bootparamd – boot parameter server

形式

```
/usr/sbin/rpc.bootparamd [-d]
```

機能説明

`rpc.bootparamd` is a server process that provides information from a `bootparams` database to diskless clients at boot time. See `bootparams(4)`

The source for the `bootparams` database is determined by the `nsswitch.conf(4)` file (on the machine running the `rpc.bootparamd` process).

オプション

The following options are supported:

`-d` Display debugging information.

ファイル

`/etc/bootparams` boot parameter data base

`/etc/nsswitch.conf` configuration file for the name-service switch

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/network

関連項目

`svcs(1)`,`svcadm(1M)`, `bootparams(4)`, `nsswitch.conf(4)`, `attributes(5)`, `smf(5)`

注意事項

A diskless client requires service from at least one `rpc.bootparamd` process running on a server that is on the same IP subnetwork as the diskless client.

Some routines that compare hostnames use case-sensitive string comparisons; some do not. If an incoming request fails, verify that the case of the hostname in the file to be parsed matches the case of the hostname called for, and attempt the request again.

The `rpc.bootparamd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/bootparams
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

rpcinfo – report RPC information

形式

```
rpcinfo [-m | -s] [host]
rpcinfo -p [host]
rpcinfo -T transport host prognum [versnum]
rpcinfo -l [-T transport] host prognum versnum
rpcinfo [-n portnum] -u host prognum [versnum]
rpcinfo [-n portnum] -t host prognum [versnum]
rpcinfo -a serv_address -T transport prognum [versnum]
rpcinfo -b [-T transport] prognum versnum
rpcinfo -d [-T transport] prognum versnum
```

機能説明

`rpcinfo` makes an RPC call to an RPC server and reports what it finds.

In the first synopsis, `rpcinfo` lists all the registered RPC services with `rpcbind` on *host*. If *host* is not specified, the local host is the default. If `-s` is used, the information is displayed in a concise format.

In the second synopsis, `rpcinfo` lists all the RPC services registered with `rpcbind`, version 2. Note that the format of the information is different in the first and the second synopsis. This is because the second synopsis is an older protocol used to collect the information displayed (version 2 of the `rpcbind` protocol).

The third synopsis makes an RPC call to procedure 0 of *prognum* and *versnum* on the specified *host* and reports whether a response was received. *transport* is the transport which has to be used for contacting the given service. The remote address of the service is obtained by making a call to the remote `rpcbind`.

The *prognum* argument is a number that represents an RPC program number (see `rpc(4)`).

If a *versnum* is specified, `rpcinfo` attempts to call that version of the specified *prognum*. Otherwise, `rpcinfo` attempts to find all the registered version numbers for the specified *prognum* by calling version 0, which is presumed not to exist; if it does exist, `rpcinfo` attempts to obtain this information by calling an extremely high version number instead, and attempts to call each registered version. Note that the version number is required for `-b` and `-d` options.

The `EXAMPLES` section describe other ways of using `rpcinfo`.

オプション

`-T` *transport*

Specify the transport on which the service is required. If this option is not specified, `rpcinfo` uses the transport specified in the `NETPATH` environment variable, or if that is unset or `NULL`, the transport in the `netconfig(4)` database is used. This is a generic option, and can be used in conjunction with other options as shown in the `SYNOPSIS`.

`-a` *serv_address*	Use *serv_address* as the (universal) address for the service on *transport* to `ping` procedure 0 of the specified *prognum* and report whether a response was received. The `-T` option is required with the `-a` option. If *versnum* is not specified, `rpcinfo` tries to ping all available version numbers for that program number. This option avoids calls to remote `rpcbind` to find the address of the service. The *serv_address* is specified in universal address format of the given transport.
`-b`	Make an RPC broadcast to procedure 0 of the specified *prognum* and *versnum* and report all hosts that respond. If *transport* is specified, it broadcasts its request only on the specified transport. If broadcasting is not supported by any transport, an error message is printed. Use of broadcasting should be limited because of the potential for adverse effect on other systems.
`-d`	Delete registration for the RPC service of the specified *prognum* and *versnum*. If *transport* is specified, unregister the service on only that transport, otherwise unregister the service on all the transports on which it was registered. Only the owner of a service can delete a registration, except the superuser, who can delete any service.
`-l`	Display a list of entries with a given *prognum* and *versnum* on the specified *host*. Entries are returned for all transports in the same protocol family as that used to contact the remote `rpcbind`.
`-m`	Display a table of statistics of `rpcbind` operations on the given *host*. The table shows statistics for each version of `rpcbind` (versions 2, 3 and 4), giving the number of times each procedure was requested and successfully serviced, the number and type of remote call requests that were made, and information about RPC address lookups that were handled. This is useful for monitoring RPC activities on *host*.
`-n` *portnum*	Use *portnum* as the port number for the `-t` and `-u` options instead of the port number given by `rpcbind`. Use of this option avoids a call to the remote `rpcbind` to find out the address of the service. This option is made obsolete by the `-a` option.
`-p`	Probe `rpcbind` on *host* using version 2 of the `rpcbind` protocol, and display a list of all registered RPC programs. If *host* is not specified, it defaults to the local host. This option is not useful for IPv6; use `-s` (see below) instead. Note that version 2 of the `rpcbind` protocol was previously known as the portmapper protocol.
`-s`	Display a concise list of all registered RPC programs on *host*. If *host* is not specified, it defaults to the local host.

`-t` Make an RPC call to procedure 0 of *prognum* on the specified *host* using TCP, and report whether a response was received. This option is made obsolete by the `-T` option as shown in the third synopsis.

`-u` Make an RPC call to procedure 0 of *prognum* on the specified *host* using UDP, and report whether a response was received. This option is made obsolete by the `-T` option as shown in the third synopsis.

使用例

例 1 RPC services.

To show all of the RPC services registered on the local machine use:

```
example% rpcinfo
```

To show all of the RPC services registered with `rpcbind` on the machine named `klaxon` use:

```
example% rpcinfo klaxon
```

The information displayed by the above commands can be quite lengthy. Use the `-s` option to display a more concise list:

```
example% rpcinfo -s klaxon
```

program	vrsn	netid(s)	service	owner
100000	2,3,4	tcp,udp,ticlts,ticots,ticotsord	rpcbind	superuser
100008	1	ticotsord,ticots,ticlts,udp,tcp	walld	superuser
100002	2,1	ticotsord,ticots,ticlts,udp,tcp	rusersd	superuser
100001	2,3,4	ticotsord,ticots,tcp,ticlts,udp	rstatd	superuser
100012	1	ticotsord,ticots,ticlts,udp,tcp	sprayd	superuser
100007	3	ticotsord,ticots,ticlts,udp,tcp	ypbind	superuser
100029	1	ticotsord,ticots,ticlts	keyserv	superuser
100078	4	ticotsord,ticots,ticlts	-	superuser
100024	1	ticotsord,ticots,ticlts,udp,tcp	status	superuser
100021	2,1	ticotsord,ticots,ticlts,udp,tcp	nlockmgr	superuser
100020	1	ticotsord,ticots,ticlts,udp,tcp	llockmgr	superuser

To show whether the RPC service with program number *prognum* and version *versnum* is registered on the machine named `klaxon` for the transport TCP use:

```
example% rpcinfo -T tcp klaxon prognum versnum
```

To show all RPC services registered with version 2 of the `rpcbind` protocol on the local machine use:

```
example% rpcinfo -p
```

To delete the registration for version 1 of the `walld` (program number `100008`) service for all transports use:

```
example# rpcinfo -d 100008 1
```

or

```
example# rpcinfo -d walld 1
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`rpcbind(1M)`, `rpc(3NSL)`, `netconfig(4)`, `rpc(4)`, `attributes(5)`

名前

rpc.mdcommd – multi-node disk set services

形式

```
/usr/sbin/rpc.mdcommd
```

機能説明

rpc.mdcommd is an rpc(4) daemon that functions as a server process. rpc.mdcommd manages communication among hosts participating in a multi-node disk set configuration.

rpc.mdcommd is invoked by inetd(1M).

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

svcs(1), inetadm(1M), inetd(1M), metaset(1M), svcadm(1M), rpc(3NSL), rpc(4), services(4), attributes(5), smf(5)

『Solaris ボリュームマネージャの管理』

注意事項

The rpc.mdcommd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rpc/mdcomm
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

rpc.metad – リモートメタセットサービス

形式

/usr/sbin/rpc.metad

機能説明

rpc.metad は、メタデバイスのディスクセット情報のローカルコピーを管理するのに使用される rpc(4) デーモン (サーバープロセスとして機能する) です。rpc.metad デーモンは inetadm(1M) によって制御されます。

終了ステータス

次の終了値が返されます。

0　正常終了。

>0　エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/svm

関連項目

svcs(1), inetadm(1M), inetd(1M), metaset(1M), rpc.metamhd(1M), svcadm(1M), rpc(3NSL), services(4), attributes(5), smf(5)

『Solaris ボリュームマネージャの管理』

注意事項

rpc.metad サービスを管理するには、サービス管理機能 smf(5) を使用します。このとき、次のサービス識別子を使用します。

```
svc:/network/rpc/meta:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。このサービスを開始または再開始する責任は inetd(1M) に委託されています。このサービスの構成を変更したり、構成情報を表示したりするには、inetadm(1M) を使用します。サービスのステータスは svcs(1) コマンドを使用して照会できます。

名前

rpc.metamedd – remote mediator services

形式

`/usr/sbin/rpc.metamedd`

機能説明

`rpc.metamedd` is an `rpc(4)` server which is used to manage mediator information for use in 2–string HA configurations. The `rpc.metamedd` daemon is controlled by `inetadm(1M)`.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	storage/svm
Interface Stability	Committed

関連項目

`svcs(1)`, `inetadm(1M)`, `inetd(1M)`, `svcadm(1M)`, `rpc(4)`, `services(4)`, `attributes(5)`, `smf(5)`

Sun Cluster documentation, 『Solaris ボリュームマネージャの管理』

注意事項

The `rpc.metamedd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/metamed:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

rpc.metamhd – リモートマルチホストディスクサービス

形式

/usr/sbin/rpc.metamhd

機能説明

rpc.metamhd は、複数のホストに接続されているディスクを管理するのに使用される rpc(4) デーモン (サーバープロセスとして機能する) です。rpc.metamhd デーモンは inetadm(1M) によって制御されます。

終了ステータス

次の終了値が返されます。

0　正常終了。

>0　エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/svm

関連項目

svcs(1), inetadm(1M), inetd(1M), metaset(1M), rpc.metad(1M), svcadm(1M), rpc(3NSL), services(4), attributes(5), smf(5)

『Solaris ボリュームマネージャの管理』

注意事項

rpc.metamhd サービスを管理するには、サービス管理機能 smf(5) を使用します。このとき、次のサービス識別子を使用します。

```
svc:/network/rpc/metamh:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。このサービスを開始または再開始する責任は inetd(1M) に委託されています。このサービスの構成を変更したり、構成情報を表示したりするには、inetadm(1M) を使用します。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

名前

rpc.rexd, rexd – RPC-based remote execution server

形式

```
/usr/sbin/rpc.rexd [-s]
```

機能説明

`rpc.rexd` is the Sun RPC server for remote program execution. This daemon is started by `inetd(1M)` whenever a remote execution request is made.

For non-interactive programs, the standard file descriptors are connected directly to TCP connections. Interactive programs involve pseudo-terminals, in a fashion that is similar to the login sessions provided by `rlogin(1)`. This daemon may use NFS to mount file systems specified in the remote execution request.

There is a `10240` byte limit for arguments to be encoded and passed from the sending to the receiving system.

オプション

The following option is supported:

`-s` Secure. When specified, requests must have valid DES credentials. If the request does not have a DES credential it is rejected. The default publickey credential is rejected. Only newer `on(1)` commands send DES credentials.

If access is denied with an authentication error, you may have to set your publickey with the `chkey(1)` command.

Specifying the `-s` option without presenting secure credentials will result in an error message: `Unix too weak auth (DesONly)!`

セキュリティー

`rpc.rexd` uses `pam(3PAM)` for account and session management. The PAM configuration policy, listed through `/etc/pam.conf`, specifies the modules to be used for `rpc.rexd`. Here is a partial `pam.conf` file with `rpc.rexd` entries for account and session management using the UNIX module.

```
rpc.rexd   account requisite       pam_roles.so.1
rpc.rexd   account required        pam_projects.so.1
rpc.rexd   account required        pam_unix_account.so.1

rpc.rexd      session required      pam_unix_session.so.1
```

If there are no entries for the `rpc.rexd` service, the entries for the “other” service will be used. `rpc.rexd` uses the `getpwuid()` call to determine whether the given user is a legal user.

ファイル

`/dev/pts`*n*	Pseudo-terminals used for interactive mode
`/etc/passwd`	Authorized users
`/tmp_rex/rexd??????`	Temporary mount points for remote file systems

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

chkey(1), on(1), rlogin(1), svcs(1), inetadm(1M), inetd(1M), svcadm(1M), pam(3PAM), pam.conf(4), publickey(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5)

診断

Diagnostic messages are normally printed on the console, and returned to the requestor.

注意事項

Root cannot execute commands using rexd client programs such as on(1).

The rpc.rexd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rpc/rex:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

rpc.rstatd, rstatd – kernel statistics server

形式

```
/usr/lib/netsvc/rstat/rpc.rstatd
```

機能説明

`rpc.rstatd` is a server which returns performance statistics obtained from the kernel. `rup(1)` uses `rpc.rstatd` to collect the `uptime` information that it displays.

`rpc.rstatd` is an RPC service.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

`rup(1)`, `svcs(1)`, `inetadm(1M)`, `inetd(1M)`, `svcadm(1M)`, `services(4)`, `attributes(5)`, `smf(5)`

注意事項

The `rpc.rstatd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/rstat:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

rpc.rusersd, rusersd – network username server

形式

```
/usr/lib/netsvc/rusers/rpc.rusersd
```

機能説明

`rpc.rusersd` is a server that returns a list of users on the host. The `rpc.rusersd` daemon may be started by `inetd(1M)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

`svcs(1)`, `inetadm(1M)`, `inetd(1M)`, `svcadm(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `rpc.rusersd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/rusers:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

rpc.rwalld, rwalld – network rwall server

形式

```
/usr/lib/netsvc/rwall/rpc.rwalld
```

機能説明

`rpc.rwalld` is a server that handles `rwall(1M)` requests. It is implemented by calling `wall(1M)` on all the appropriate network machines. The `rpc.rwalld` daemon may be started by `inetd(1M)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

`svcs(1)`, `inetadm(1M)`, `inetd(1M)`, `rwall(1M)`, `svcadm(1M)`, `wall(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `rpc.rwalld` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/wall:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

rpc.smserverd – removable media device server

形式

```
/usr/lib/smedia/rpc.smserverd
```

機能説明

rpc.smserverd is a server that handles requests from client applications, such as volume management software, for access to removable media devices. In addition to volume management software, rmformat(1) and the CDE Filemanager (when performing removable media operations) are rpc.smserverd clients. The rpc.smserverd daemon is started by inetd(1M) when a client makes a call to a Solaris-internal library to access a SCSI, IDE, or USB device. The daemon is not started if a client attempts to access a PCMCIA device. Once started, the daemon remains active until such time as it is idle for three minutes or more.

The rpc.smserverd daemon is provided for the exclusive use of the client applications mentioned above. It has no external, customer-accessible interfaces, including no configuration file.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/storage/removable-media

関連項目

svcs(1), inetadm(1M), inetd(1M), svcadm(1M), attributes(5), smf(5)

注意事項

The rpc.smserverd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/rpc/smserver
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

名前

rpc.sprayd, sprayd – spray server

形式

```
/usr/lib/netsvc/spray/rpc.sprayd
```

機能説明

`rpc.sprayd` is a server that records the packets sent by `spray(1M)`. The `rpc.sprayd` daemon may be started by `inetd(1M)`.

The service provided by `rpc.sprayd` is not useful as a networking benchmark as it uses unreliable connectionless transports, (udp for example). It can report a large number of packets dropped when the drops were caused by the program sending packets faster than they can be buffered locally (before the packets get to the network medium).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

関連項目

`svcs(1)`, `inetadm(1M)`, `inetd(1M)`, `spray(1M)`, `svcadm(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `rpc.sprayd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/rpc/spray:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. Responsibility for initiating and restarting this service is delegated to `inetd(1M)`. Use `inetadm(1M)` to make configuration changes and to view configuration information for this service. The service's status can be queried using the `svcs(1)` command.

名前

rpc.yppasswdd, yppasswdd – server for modifying NIS password file

形式

```
/usr/lib/netsvc/yp/rpc.yppasswdd [-D directory]
     [-nogecos] [-noshell] [-nopw]
     [-m argument1 argument2...]
```

```
/usr/lib/netsvc/yp/rpc.yppasswdd
     [passwordfile [adjunctfile]]
     [-nogecos] [-noshell] [-nopw]
     [-m argument1 argument2...]
```

機能説明

`rpc.yppasswdd` is a server that handles password change requests from `yppasswd(1)`. It changes a password entry in the `passwd`, `shadow`, and `security/passwd.adjunct` files. The `passwd` and `shadow` files provide the basis for the `passwd.byname` and `passwd.byuid` maps. The `passwd.adjunct` file provides the basis for the `passwd.adjunct.byname` and `passwd.adjunct.byuid` maps. Entries in the `passwd`, `shadow` or `passwd.adjunct` files are changed only if the password presented by `yppasswd(1)` matches the encrypted password of the entry. All password files are located in the `PWDIR` directory.

If the `-D` option is given, the `passwd`, `shadow`, or `passwd.adjunct` files are placed under the directory path that is the argument to `-D`.

If the `-noshell`, `-nogecos` or `-nopw` options are given, these fields cannot be changed remotely using `chfn`, `chsh`, or `passwd(1)`.

If the `-m` option is given, a `make(1S)` is performed in `/var/yp` after any of the `passwd`, `shadow`, or `passwd.adjunct` files are modified. All arguments following the flag are passed to `make`.

The second of the listed syntaxes is provided only for backward compatibility. If the second syntax is used, the `passwordfile` is the full pathname of the password file and `adjunctfile` is the full pathname of the optional `passwd.adjunct` file. If a shadow file is found in the same directory as `passwordfile`, the `shadowfile` is used as described above. Use of this syntax and the discovery of a `shadowfile` file generates diagnostic output. The daemon, however, starts normally.

The first and second syntaxes are mutually exclusive. You cannot specify the full pathname of the `passwd`, `passwd.adjunct` files and use the `-D` option at the same time.

The daemon is started automatically on the master server of the passwd map by `ypstart(1M)`, which is invoked at boot time by the `svcs:/network/nis/server:default` service.

The server does not insist on the presence of a `shadow` file unless there is no `-D` option present or the directory named with the `-D` option is `/etc`. In addition, a `passwd.adjunct` file is not necessary. If the `-D` option is given, the server attempts to find a `passwd.adjunct` file in the `security` subdirectory of the named directory. For example, in the presence of `-D /var/yp` the server checks for a `/var/yp/security/passwd.adjunct` file.

If only a passwd file exists, then the encrypted password is expected in the second field. If both a passwd and a passwd.adjunct file exist, the encrypted password is expected in the second field of the adjunct file with *##username* in the second field of the passwd file. If all three files are in use, the encrypted password is expected in the shadow file. Any deviation causes a password update to fail.

If you remove or add a shadow or passwd.adjunct file after rpc.yppasswdd has started, you must stop and restart the daemon to enable it to recognize the change. See ypstart(1M) for information on restarting the daemon.

The rpc.yppasswdd daemon considers a shell that has a name that begins with 'r' to be a restricted shell. By default, the daemon does not check whether a shell begins with an 'r'. However, you can tell it to do so by uncommenting the check_restricted_shell_name=1 line in /etc/default/yppasswdd. The result will be to restrict a user's ability to change from his default shell. See yppasswdd(4).

On start up, yppasswdd checks for the existence of a NIS to LDAP (N2L) configuration file, /var/yp/NISLDAPmapping. If the configuration file is present, the daemon runs in N2L mode. If the file is not present, yppasswdd runs in traditional, non-N2L mode.

In N2L mode, changes are written directly to the Directory Information Tree (DIT). If the changes are written successfully, the NIS map is updated. The NIS source files, passwd, shadow, and passwd.adjunct, for example, are not updated. Thus, in N2L mode, the -D option is meaningless. In N2L mode, yppasswdd propagates changes by calling yppush(1M) instead of ypmake(1M). The -m option is thus unused.

During an NIS-to-LDAP transition, the yppasswdd daemon uses the N2L-specific map, ageing.byname, to read and write password aging information to the DIT. If you are not using password aging, then the ageing.byname mapping is ignored.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis

関連項目

svcs(1), make(1S), passwd(1), yppasswd(1), inetd(1M), svcadm(1M), ypmake(1M), yppush(1M), ypstart(1M), NISLDAPmapping(4), passwd(4), shadow(4), ypfiles(4), yppasswdd(4), ypserv(4), attributes(5), smf(5)

注意事項

If make has not been installed and the -m option is given, the daemon outputs a warning and proceeds, effectively ignoring the -m flag.

When using the -D option, you should make sure that the PWDIR of the /var/yp/Makefile is set accordingly.

The second listed syntax is supplied only for backward compatibility and might be removed in a future release of this daemon.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications PLC, and cannot be used without permission.

The NIS server service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svcs:/network/nis/server:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

rpc.ypupdated, ypupdated – server for changing NIS information

形式

```
/usr/lib/netsvc/yp/rpc.ypupdated [-is]
```

機能説明

ypupdated is a daemon that updates information in the Network Information Service (NIS). ypupdated consults the updaters(4) file in the /var/yp directory to determine which NIS maps should be updated and how to change them.

By default, the daemon requires the most secure method of authentication available to it, either DES (secure) or UNIX (insecure).

On start up, ypupdated checks for the existence of a NIS to LDAP (N2L) configuration file, /var/yp/NISLDAPmapping. If the file is present, ypupdated generates an informational message and exits. ypupdated is not supported in N2L mode.

オプション

-i　Accept RPC calls with the insecure AUTH_UNIX credentials. This allows programmatic updating of the NIS maps in all networks.

-s　Accept only calls authenticated using the secure RPC mechanism (AUTH_DES authentication). This disables programmatic updating of the NIS maps unless the network supports these calls.

ファイル

/var/yp/updaters	Configuration file for rpc.updated command.
/var/yp/NISLDAPmapping	Configuration file for N2L

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis
Interface Stability	Committed

関連項目

keyserv(1M), updaters(4), NISLDAPmapping(4), attributes(5)

『Oracle Solaris Administration: Naming and Directory Services』

注意事項

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two services remains the same. Only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications PLC, and it must not be used without permission.

名前

rquotad – remote quota server

形式

/usr/lib/nfs/rquotad

機能説明

rquotad is an rpc(4) server which returns quotas for a user of a local file system which is mounted by a remote machine over the NFS. The results are used by quota(1M) to display user quotas for remote file systems. The rquotad daemon is normally invoked by inetd(1M).

使用法

See largefile(5) for the description of the behavior of rquotad when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

ファイル

quotas　quota file at a UFS file system root

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

関連項目

svcs(1), automountd(1M), inetadm(1M), inetd(1M), mount_nfs(1M), quota(1M), share_nfs(1M), svcadm(1M), rpc(4), services(4), attributes(5), largefile(5), smf(5)

『Oracle Solaris 11 システムのインストール』

注意事項

The rquotad service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/nfs/rquota
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). Responsibility for initiating and restarting this service is delegated to inetd(1M). Use inetadm(1M) to make configuration changes and to view configuration information for this service. The service's status can be queried using the svcs(1) command.

If it is disabled, it is enabled by mount_nfs(1M), share_nfs(1M), and automountd(1M) unless its application/auto_enable property is set to false.

名前

rsh, restricted_shell – 制限付きシェルのコマンドインタプリタ

形式

`/usr/lib/rsh [-acefhiknprstuvx] [argument]...`

機能説明

rsh は、標準のコマンドインタプリタである sh と比べて機能の一部が制限されており、ログインが許される実行環境も、sh の環境と比べて制限されています。機能の詳細や使用方法に関しては、sh(1) の説明を参照してください。

シェルは、呼び出されると環境変数 SHELL を調べます。この環境変数が存在していて、その値のファイル名部分が rsh であれば、シェルは制限付きシェルとなります。

rsh の機能は、以下の動作ができない点を除き sh と同一です。

- ディレクトリの変更 (cd(1) を参照)
- $PATH の値の設定
- / を含むパスまたはコマンド名の指定
- 出力先のリダイレクト (> および >>)

これらの制限は、*.profile* の解釈後に有効となります。

制限付きシェルは、次のいずれかの方法で呼び出せます。

1. /etc/passwd ファイルの最後のエントリのファイル名部分を rsh と記述する (passwd(4) を参照)
2. 環境変数 SHELL が存在していて、その値のファイル名の部分が rsh である。環境変数 SHELL は、 .login ファイル内に設定されている必要がある
3. シェルの呼び出し時に、 引数 0 のファイル名の部分が rsh である
4. シェルを -r オプション付きで呼び出す

実行するコマンドがシェル手続きである場合は、rsh は sh を呼び出して、コマンドを実行します。したがって、利用できるコマンドの種類には制限はありますが、一般ユーザーは標準シェルの全機能を利用できるシェル手続きを使用することができます。このスキーマは、一般ユーザーが 同じディレクトリにおいて書き込み権と実行権を持っていないことを想定しています。

.profile の作成者 (profile(4) を参照) が、確実な設定処理を実行してユーザーを適切なディレクトリ (おそらく、ログインディレクトリではない) に置くことにより、ユーザーの動作を完全に制御できるという点が、これらの規約の実際の効果となります。

システム管理者は、制限付きシェルで安全に起動できるコマンドのディレクトリ (つまり /usr/rbin) を設定することがよくあります。 システムによっては、制限付きエディタ red を提供するものもあります。

終了ステータス

構文エラーなどのエラーを検出すると、シェルは 0 以外の終了ステータスを返します。シェルを非対話型で使用している場合、シェルファイルの実行は中止されます。それ以外の場合は、シェルは最後に実行されたコマンドの 終了ステータスを返します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

Intro(1), cd(1), login(1), rsh(1), sh(1), exec(2), passwd(4), profile(4), attributes(5)

注意事項

制限付きシェル /usr/lib/rsh を、リモートシェル /usr/bin/rsh と混同しないように注意してください。リモートシェルに関しては rsh(1) を参照してください。

名前

rtc – provide all real-time clock and UTC-lag management

形式

`/usr/sbin/rtc [-c] [-z zone-name]`

機能説明

On x86 systems, the `rtc` command reconciles the difference in the way that time is established between UNIX and MS-DOS systems. UNIX systems utilize Universal Coordinated Time (UTC), while MS-DOS systems utilize local time.

Without arguments, `rtc` displays the currently configured time zone string. The currently configured time zone string is based on what was last recorded by `rtc -z` *zone-name*.

The `rtc` command is not normally run from a shell prompt; it is generally invoked by the system. Commands such as `date(1)` and `rdate(1M)`, which are used to set the time on a system, invoke `/usr/sbin/rtc -c` to ensure that daylight savings time (DST) is corrected for properly.

オプション

`-c`

This option checks for DST and makes corrections if necessary. It is normally run once a day by a `cron` job.

If there is no RTC time zone or `/etc/rtc_config` file, this option will do nothing.

`-z` *zone-name*

This option, which is normally run by the system at software installation time, is used to specify the time zone in which the RTC is to be maintained. It updates the configuration file `/etc/rtc_config` with the name of the specified zone and the current UTC lag for that zone. If there is an existing `rtc_config` file, this command will update it. If not, this command will create it.

ファイル

`/etc/rtc_config`

The data file used to record the time zone and UTC lag. This file is completely managed by `/usr/sbin/rtc`, and it is read by the kernel.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Architecture	x86
Availability	system/core-os

関連項目

`date(1)`, `rdate(1M)`, `attributes(5)`

名前

rtquery – query routing daemons for their routing tables

形式

```
rtquery [-np1] [-w timeout] [-r addr] [-a secret] host...
rtquery [-t operation] host...
```

機能説明

The `rtquery` command is used to query a RIP network routing daemon, `in.routed(1M)` or `GateD`, for its routing table by sending a request or poll command. The routing information in any routing response packets returned is displayed numerically and symbolically.

By default, `rtquery` uses the request command. When the -p option is specified, `rtquery` uses the poll command, an undocumented extension to the RIP protocol supported by `GateD`. When querying `GateD`, the poll command is preferred over the request command because the response is not subject to Split Horizon and/or Poisoned Reverse, and because some versions of `GateD` do not answer the request command. `in.routed` does not answer the poll command, but recognizes requests coming from `rtquery` and so answers completely.

The `rtquery` command is also used to turn tracing on or off in `in.routed`.

オプション

The following options are supported:

`-a passwd=`*XXX*
`-a md5_passwd=`*XXX*|*KeyID*
: Causes the query to be sent with the indicated cleartext or MD5 password.

`-n`
: Displays only the numeric network and host addresses instead of both numeric and symbolic names.

`-p`
: Uses the poll command to request full routing information from `GateD`. This is an undocumented extension RIP protocol supported only by `GateD`.

`-r` *addr*
: Asks about the route to destination *addr*.

`-t` *operation*
: Changes tracing, where *operation* is one of the actions listed below. Requests from processes not running with UID 0 or on distant networks are generally ignored by the daemon except for a message in the system log. `GateD` is likely to ignore these debugging requests.

  on=*tracefile*
  : Turns tracing on, directing tracing into the specified file. That file must have been specified when the daemon was started or have the name, `/var/log/in.routed.trace`.

  more
  : Increases the debugging level.

  off
  : Turns off tracing.

dump	Dumps the daemon's routing table to the current trace file.

-w *timeout* Changes the delay for an answer from each host. By default, each host is given 15 seconds to respond.

-1 Queries using RIP version 1 instead of RIP version 2.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/routing

関連項目

in.routed(1M), route(1M), gateways(4), attributes(5), icmp(7P), inet(7P), udp(7P)

Routing Information Protocol, RIPv1, RFC 1058

Routing Information Protocol, RIPv2, RFC 2453, STD 0056

名前

runacct – run daily accounting

形式

`/usr/lib/acct/runacct [mmdd [state]]`

機能説明

`runacct` is the main daily accounting shell procedure. It is normally initiated using `cron`. `runacct` processes connect, fee, disk, and process accounting files. It also prepares summary files for `prdaily` or billing purposes. `runacct` is distributed only to source code licensees.

`runacct` takes care not to damage active accounting files or summary files in the event of errors. It records its progress by writing descriptive diagnostic messages into `active`. When an error is detected, a message is written to `/dev/console`, mail (see `mail(1)`) is sent to `root` and `adm`, and `runacct` terminates. `runacct` uses a series of lock files to protect against re-invocation. The files `lock` and `lock1` are used to prevent simultaneous invocation, and `lastdate` is used to prevent more than one invocation per day.

`runacct` breaks its processing into separate, restartable *states* using `statefile` to remember the last *state* completed. It accomplishes this by writing the *state* name into `statefile`. `runacct` then looks in `statefile` to see what it has done and to determine what to process next. *states* are executed in the following order:

`SETUP`	Move active accounting files into working files.
`WTMPFIX`	Verify integrity of `wtmpx` file, correcting date changes if necessary.
`CONNECT`	Produce connect session records in `tacct.h` format.
`PROCESS`	Convert process accounting records into `tacct.h` format.
`MERGE`	Merge the connect and process accounting records.
`FEES`	Convert output of `chargefee` into `tacct.h` format, merge with connect, and process accounting records.
`DISK`	Merge disk accounting records with connect, process, and fee accounting records.
`MERGETACCT`	Merge the daily total accounting records in `daytacct` with the summary total accounting records in `/var/adm/acct/sum/tacct`.
`CMS`	Produce command summaries.
`USEREXIT`	Any installation dependent accounting programs can be included here.
`CLEANUP`	Clean up temporary files and exit. To restart `runacct` after a failure, first check the `active` file for diagnostics, then fix any corrupted data files, such as `pacct` or `wtmpx`. The `lock`, `lock1`, and `lastdate` files must be removed before `runacct` can be restarted. The argument *mmdd* is necessary if `runacct` is being restarted. *mmdd* specifies the month and day for which `runacct` will rerun the accounting. The entry point for processing is based on the contents of `statefile`; to override this, include the desired *state* on the command line to designate where processing should begin.

使用例

例 1 Starting runacct

The following example starts runacct:

```
example% nohup runacct 2> /var/adm/acct/nite/fd2log &
```

例 2 Restarting runacct

The following example restarts runacct:

```
example% nohup runacct 0601 2>> /var/adm/acct/nite/fd2log &
```

例 3 Restarting runacct at a Specific State

The following example restarts runacct at a specific state:

```
example% nohup runacct 0601 MERGE 2>> /var/adm/acct/nite/fd2log &
```

ファイル

/var/adm/wtmpx	History of user access and administration information
/var/adm/pacct*incr*	
/var/adm/acct/nite/active	
/var/adm/acct/nite/daytacct	
/var/adm/acct/nite/lock	
/var/adm/acct/nite/lock1	
/var/adm/acct/nite/lastdate	
/var/adm/acct/nite/statefile	

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

acctcom(1), mail(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), cron(1M), fwtmp(1M), acct(2), acct.h(3HEAD), utmpx(4), attributes(5)

注意事項

It is not recommended to restart runacct in the SETUP *state*. Run SETUP manually and restart using:

runacct *mmdd* WTMPFIX

If runacct failed in the PROCESS *state*, remove the last ptacct file because it will not be complete.

The runacct command can process a maximum of

- 6000 distinct sessions

- 1000 distinct terminal lines
- 2000 distinct login names

during a single invocation of the command. If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.

Do not invoke `runacct` at the same time as `ckpacct`, as there may be a conflict if both scripts attempt to execute `turnacct switch` simultaneously.

名前

rwall – write to all users over a network

形式

`/usr/sbin/rwall` *hostname*...

`/usr/sbin/rwall -n` *netgroup*...

`/usr/sbin/rwall -h` *hostname* `-n` *netgroup*

機能説明

`rwall` reads a message from standard input until EOF. It then sends this message, preceded by the line:

```
Broadcast Message . . .
```

to all users logged in on the specified host machines. With the `-n` option, it sends to the specified network groups.

オプション

`-n` *netgroup* Send the broadcast message to the specified network groups.

`-h` *hostname* Specify the `hostname`, the name of the host machine.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

関連項目

`inetd(1M)`, `wall(1M)`, `attributes(5)`

注意事項

The timeout is fairly short to allow transmission to a large group of machines (some of which may be down) in a reasonable amount of time. Thus the message may not get through to a heavily loaded machine.

名前

sar, sa1, sa2, sadc – system activity report package

形式

```
/usr/lib/sa/sadc [t n] [ofile]

/usr/lib/sa/sa1 [t n]

/usr/lib/sa/sa2 [-aAbcdgkmpqruvwy] [-e time] [-f filename]
     [-i sec] [-s time]
```

機能説明

System activity data can be accessed at the special request of a user (see sar(1)) and automatically, on a routine basis, as described here. The operating system contains several counters that are incremented as various system actions occur. These include counters for CPU utilization, buffer usage, disk and tape I/O activity, TTY device activity, switching and system-call activity, file-access, queue activity, inter-process communications, and paging. For more general system statistics, use iostat(1M), sar(1), or vmstat(1M).

sadc and two shell procedures, sa1 and sa2, are used to sample, save, and process this data.

sadc, the data collector, samples system data *n* times, with an interval of *t* seconds between samples, and writes in binary format to *ofile* or to standard output. The sampling interval *t* should be greater than 5 seconds; otherwise, the activity of sadc itself may affect the sample. If *t* and *n* are omitted, a special record is written. This facility can be used at system boot time, when booting to a multi-user state, to mark the time at which the counters restart from zero. For example, when accounting is enabled, the svc:/system/sar:default service writes the restart mark to the daily data file using the command entry:

```
su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa‘date +%d‘"
```

The shell script sa1, a variant of sadc, is used to collect and store data in the binary file /var/adm/sa/sa*dd,* where dd is the current day. The arguments *t* and *n* cause records to be written *n* times at an interval of *t* seconds, or once if omitted. The following entries in /var/spool/cron/crontabs/sys will produce records every 20 minutes during working hours and hourly otherwise:

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8–17 * * 1–5 /usr/lib/sa/sa1
```

See crontab(1) for details.

The shell script sa2, a variant of sar, writes a daily report in the file /var/adm/sa/sar*dd.* See the OPTIONS section in sar(1) for an explanation of the various options. The following entry in /var/spool/cron/crontabs/sys will report important activities hourly during the working day:

```
5 18 * * 1–5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

ファイル

/tmp/sa.adrfl	address file
/var/adm/sa/sadd	daily data file
/var/adm/sa/sardd	daily report file

`/var/spool/cron/crontabs/sys` used for performance collection

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/accounting/legacy-accounting

関連項目

`crontab(1)`, `sar(1)`, `svcs(1)`, `timex(1)`, `iostat(1M)`, `svcadm(1M)`, `vmstat(1M)`, `attributes(5)`, `smf(5)`

『Oracle Solaris の管理: 一般的なタスク』

注意事項

The `sar` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/sar
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

sasinfo – Serial Attached SCSI HBA ポートコマンド行インタフェース

形式

```
sasinfo hba [-v] [HBA_Name]...

sasinfo hba-port [-lvy] [-a HBA_Name] [HBA_port_nam]...

sasinfo expander [-vt] [-p HBA_port_Name] [Expander_SAS_Addr]...

sasinfo target-port [-s | -v] [Target_port_SAS_Addr]...

sasinfo logical-unit | lu [-v] [device_path]...

sasinfo [-V]

sasinfo [-?]
```

機能説明

`sasinfo` ユーティリティーは、Storage Management HBA API (SM-HBA) をサポートする Serial Attached SCSI-2 (SAS-2) ホストバスアダプタ (HBA) に関する管理情報を収集するコマンド行インタフェースです。このユーティリティーは、HBA ポートの属性と、この HBA ポートに接続できるエキスパンダデバイスおよび SCSI ターゲットデバイスの属性を報告します。

`sasinfo` は、次に説明するように、一連のサブコマンドとして実装されています。

サブコマンド

`sasinfo` では次のサブコマンドがサポートされています。

`hba`
: 指定された *HBA_name* で参照される HBA に関する情報を一覧表示します。*HBA_name* が指定されていない場合は、ホスト上のすべての HBA が一覧表示されます。

`hba-port`
: 指定された *HBA_port_Name* で参照される HBA ポートに関する情報を一覧表示します。*HBA_port_Name* が指定されていない場合は、ホスト上のすべての HBA ポートが一覧表示されます。HBA ポートの構成/未構成は動的に切り換えることが可能です。そのため、ターゲットデバイスとの接続の確立に応じて HBA 上の HBA ポート数が変わります。ストレージデバイスへの接続がなければ、HBA ポートは構成されません。

`expander`
: 指定された *Expander_SAS_Addr* で参照されるエキスパンダの階層ビューを一覧表示します。引数が指定されていない場合は、ホスト上のすべての HBA ポートで確認できるすべてのエキスパンダが一覧表示されます。

`target-port`
: *Target_port_SAS_Addr* アドレスで参照されるターゲットポートに関する情報を一覧表示します。引数が指定されていない場合は、ホスト上で確認できるすべてのターゲットポートが一覧表示されます。

`logical-unit | lu`
: 指定された *device_path* で参照される論理ユニットを一覧表示します。*device_path* が指定されていない場合は、SAS Management Protocol (SMP) ターゲットポートな

ど、すべての SAS 論理ユニットが一覧表示されます。

オプション

次のオプションがサポートされています。

`-?`、`--help`
使用法に関する情報を表示します。

`-a`、`--hba` *HBA_Name*
ホスト上の HBA の *HBA_Name* から、HBA ポート情報を取得します。`-a` オプションは、`hba-port` サブコマンドでのみ使用されます。

`-l`、`--phy-linkstat`
指定された *HBA_port_Name* で参照される HBA ポート、または HBA ポートが指定されていない場合はすべての HBA ポート上の `phy` に関するリンクエラー統計情報を一覧表示します。このオプションは、`hba-port` サブコマンドでのみ使用されます。

`-p` *HBA_port_SAS_Addr*、`--port` *HBA_port_SAS_Addr*
ホスト上のローカル HBA ポートの *HBA_port_SAS_Addr* から、リモートポート情報を取得します。`-p` オプションは、`expander` サブコマンドでのみ使用されます。

`-s`、`--scsi`
表示するように要求されたターゲットポートの SCSI 属性を一覧表示します。このオプションは、`target-port` サブコマンドでのみ使用されます。

`-v`、`--verbose`
呼び出されるサブコマンドに応じて、SAS アドレスやトポロジデバイスなどのハードウェア情報に関する詳細を表示します。

`-V`、`--version`
バージョン情報を表示します。

`-y`、`--phy`
HBA_port_Name で指定された HBA ポート、または HBA ポートが指定されていない場合はすべての HBA ポートに関する `phy` 情報を一覧表示します。このオプションは、`hba-port` サブコマンドでのみ使用されます。

使用例

例 1　すべての HBA を一覧表示する

次のコマンドは、ホスト上のすべての HBA を一覧表示します。

```
# sasinfo hba
HBA Name: SUNW-pmcs-0
HBA Name: SUNW-pmcs-1
```

例 2　すべての HBA を詳細とともに一覧表示する

次のコマンドは、ホスト上のすべての HBA を、関連する詳細とともに一覧表示します。それぞれの HBA には、2 つの HBA ポートが構成されており、それぞれの HBA ポートはストレージデバイスに接続されています。

例 2 すべての HBA を詳細とともに一覧表示する (続き)

```
# sasinfo hba -v
HBA Name: SUNW-pmcs-0
    Manufacturer: sun
    Model: SAS Gen-2
    Firmware Version: 1.1
    FCode/BIOS Version: 1.1
    Serial Number: 111-11111
    Driver Name: smvsl
    Driver Version: 1.1
    Number of HBA Ports: 2
HBA Name: SUNW-pmcs-1
    Manufacturer: sun
    Model: SAS Gen-2
    Firmware Version: 1.1
    FCode/BIOS Version: 1.1
    Serial Number: 111-11111
    Driver Name: smvsl
    Driver Version: 1.1
    Number of HBA Ports: 2
```

例 3 すべての HBA ポートを一覧表示する

次のコマンドは、ホスト上のすべての HBA ポートを一覧表示します。

```
# sasinfo hba-port
HBA Name: SUNW-pmcs-0
    HBA Port Name: /dev/cfg/c1
    HBA Port Name: /dev/cfg/c2
HBA Name: SUNW-pmcs-1
    HBA Port Name: /dev/cfg/c3
    HBA Port Name: /dev/cfg/c4
```

例 4 すべての HBA ポートを詳細とともに一覧表示する

次のコマンドは、すべての HBA ポートを、付随する詳細とともに一覧表示します。

```
# sasinfo hba-port -v
HBA Name: SUNW-pmcs-0
    HBA Port Name: /dev/cfg/c1
        Type: sas-device
        State: online
        Local SAS Address: 5000c50000d756aa
        Attached Port SAS Address: 50800201a5a502bf
        Number of Phys: 4
    HBA Port Name: /dev/cfg/c25000c50000d756cc
        Type: sas-device
        State: online
```

例 4　すべての HBA ポートを詳細とともに一覧表示する　　*(続き)*

```
        Local SAS Address: 5000c50000d756aa
        Attached Port SAS Address: 50800201a5a503bf
        Number of Phys: 4
HBA Name: SUNW-pmcs-1
    HBA Port Name: /dev/cfg/c3
        Type: sas-device
        State: online
        Local SAS Address: 5000c50000d756cc
        Attached Port SAS Address: 50800201a5a504bf
        Number of Phys: 4
    HBA Port Name: /dev/cfg/c4
        Type: sas-device
        State: online
        Local SAS Address: 5000c50000d756cc
        Attached Port SAS Address: 50800201a5a505bf
        Number of Phys: 4
```

例 5　すべての HBA ポートに関する phy 情報を一覧表示する

次のコマンドは、すべての HBA ポートに関する phy 情報を一覧表示します。

```
# sasinfo hba-port -y
HBA Name: SUNW-pmcs-0
    HBA Port Name: /dev/cfg/c1
        Phy Information:
         Identifier: 0
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 1
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 2
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 3
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
    HBA Port Name: /dev/cfg/c2
        Phy Information:
```

例 5　すべての HBA ポートに関する phy 情報を一覧表示する　　(続き)

```
         Identifier: 4
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 5
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
        Identifier: 6
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
        Identifier: 7
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
HBA Name: SUNW-pmcs-1
    HBA Port Name: /dev/cfg/c3
        Phy Information:
         Identifier: 0
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 1
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 2
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
         Identifier: 3
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
    HBA Port Name: /dev/cfg/c4
        Phy Information:
         Identifier: 4
```

例 5　すべての HBA ポートに関する phy 情報を一覧表示する　　*(続き)*

```
          State: enabled
          HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
          ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
          NegotiatedLinkRate: 3Gbit
        Identifier: 5
          State: enabled
          HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
          ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
          NegotiatedLinkRate: 3Gbit
        Identifier: 6
          State: enabled
          HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
          ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
          NegotiatedLinkRate: 3Gbit
        Identifier: 7
          State: enabled
          HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
          ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
          NegotiatedLinkRate: 3Gbit
```

例 6　特定の HBA ポートに関する phy リンクエラー統計情報を一覧表示する

次のコマンドは、特定のポートに関する phy リンクエラー統計情報を一覧表示します。

```
# sasinfo hba-port -ly /dev/cfg/c1
HBA Name: SUNW-pmcs-0
  HBA Port Name: /dev/cfg/c1
    Phy Information:
      Identifier: 0
       State: enabled
         HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
         ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
         NegotiatedLinkRate: 3Gbit
         Link Error Statistics:
           Invalid Dword: 0
           Running Disparity Error: 0
           Loss of Dword Sync: 0
           Reset Problem: 0
      Identifier: 1
         State: enabled
         HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
         ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
         NegotiatedLinkRate: 3Gbit
         Link Error Statistics:
           Invalid Dword: 0
           Running Disparity Error: 0
```

例 6　特定の HBA ポートに関する phy リンクエラー統計情報を一覧表示する　(続き)

```
            Loss of Dword Sync: 0
            Reset Problem: 0
        Identifier: 2
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
           Link Error Statistics:
             Invalid Dword: 0
             Running Disparity Error: 0
             Loss of Dword Sync: 0
             Reset Problem: 0
        Identifier: 3
           State: enabled
           HardwareLinkRate(Min/Max): 1.5Gbit/3Gbit
           ProgrammedLinkRate(Min/Max): 1.5Gbit/3Gbit
           NegotiatedLinkRate: 3Gbit
           Link Error Statistics:
             Invalid Dword: 0
             Running Disparity Error: 0
             Loss of Dword Sync: 0
             Reset Problem: 0
```

例 7　HBA ポートに接続されているエキスパンダを一覧表示する

次のコマンドは、指定された HBA ポートから確認できるすべてのエキスパンダを一覧表示します。

```
# sasinfo expander -p /dev/cfg/c1
HBA Name: SUNW-pmcs-0
  HBA Port Name: /dev/cfg/c1
    Expander SAS Address(Tier 1): 50800201a5a502bf
       Expander SAS Address(Tier 2): 50800201a5a50233
       Expander SAS Address(Tier 2); 5000c5000d2da812
    Expander SAS Address(Tier 1): 50800201a5a503bf
       Expander SAS Address(Tier 2): 50800201a5a502d2
       Expander SAS Address(Tier 2); 5000c5000d2da823
```

例 8　エキスパンダに関する詳細情報を一覧表示する

次のコマンドは、指定された HBA ポートから確認できるすべてのエキスパンダに関する詳細情報を一覧表示します。

```
# sasinfo expander -v -p /dev/cfg/c1
HBA Name: SUNW-pmcs-0
  HBA Port Name: /dev/cfg/c1
    Expander SAS Address(Tier 1): 50800201a5a502bf
      OS Device Name: /dev/smp/expd0
```

例 8 エキスパンダに関する詳細情報を一覧表示する (続き)

```
      State: online
        Expander SAS Address(Tier 2): 50800201a5a50233
          OS Device Name: /dev/smp/expd2
          State: online
        Expander SAS Address(Tier 2); 5000c5000d2da812
          OS Device Name: /dev/smp/expd3
          State: online
    Expander SAS Address(Tier 1): 50800201a5a503bf
      OS Device Name: /dev/smp/expd1
      State: online
        Expander SAS Address(Tier 2): 50800201a5a502d2
          OS Device Name: /dev/smp/expd3
          State: online
        Expander SAS Address(Tier 2); 5000c5000d2da823
          OS Device Name: /dev/smp/expd4
          State: online
```

例 9 エキスパンダに接続されているターゲットポートを一覧表示する

次のコマンドは、指定された HBA ポートに接続されているエキスパンダに接続されているすべてのターゲットポートを一覧表示します。

```
# sasinfo expander -t -p /dev/cfg/c1
HBA Name: SUNW-pmcs-0
  HBA Port Name: /dev/cfg/c1
     Expander SAS Address(Tier 1): 50800201a5a502bf
       Target Port SAS Address: 50800201a5a504f1
          Expander SAS Address(Tier 2): 50800201a5a50233
            Target Port SAS Address: 50800201a5a502be
            Target Port SAS Address: 5000c5000d2da8b2
          Expander SAS Address(Tier 2); 5000c5000d2da812
            Target Port SAS Address: 50800201a5a502be
            Target Port SAS Address: 50800201a5a508b2
     Expander SAS Address(Tier 1): 50800201a5a503bf
       Target Port SAS Address: 50800201a5a50421
          Expander SAS Address(Tier 2): 50800201a5a502d2
            Target Port SAS Address: 50800201a5a503be
            Target Port SAS Address: 5000c5000d2da7be
          Expander SAS Address(Tier 2); 5000c5000d2da823
            Target Port SAS Address: 50800201a5a503be
            Target Port SAS Address: 5000c5000d2da7be
```

例 10 ターゲットポート情報を一覧表示する

次のコマンドは、ホスト上で検出されたすべてのターゲットポートを一覧表示します。

例 10 ターゲットポート情報を一覧表示する (続き)

```
# sasinfo target-port
Target Port SAS Address: 5000c5000bae4009
Target Port SAS Address: 5000c5000baef4b1
Target Port SAS Address: 5000c5000bae3fe1
Target Port SAS Address: 5000c5000bae49d9
Target Port SAS Address: 5000c5000bae36c5
```

例 11 ターゲットポート情報をトポロジ詳細とともに一覧表示する

次のコマンドは、すべてのターゲットポートを、それらの接続先の HBA ポートおよびエキスパンダとともに一覧表示します。

```
# sasinfo target-port -v
Target Port SAS Address: 5000c5000bae4009
  Type: SAS Device
  HBA Port Name: /dev/cfg/c7
    Expander Device SAS Address: 500e004aaaaaaa3f
  HBA Port Name: /dev/cfg/c5
    Expander Device SAS Address: 500e004aaaaaaa3f
Target Port SAS Address: 5000c5000baef4b1
  Type: SAS Device
  HBA Port Name: /dev/cfg/c7
    Expander Device SAS Address: 500e004aaaaaaa3f
  HBA Port Name: /dev/cfg/c5
    Expander Device SAS Address: 500e004aaaaaaa3f
Target Port SAS Address: 5000c5000bae3fe1
  Type: SAS Device
  HBA Port Name: /dev/cfg/c7
    Expander Device SAS Address: 500e004aaaaaaa3f
  HBA Port Name: /dev/cfg/c5
    Expander Device SAS Address: 500e004aaaaaaa3f
Target Port SAS Address: 5000c5000bae49d9
  Type: SAS Device
  HBA Port Name: /dev/cfg/c7
    Expander Device SAS Address: 500e004aaaaaaa3f
  HBA Port Name: /dev/cfg/c5
    Expander Device SAS Address: 500e004aaaaaaa3f
Target Port SAS Address: 5000c5000bae36c5
  Type: SAS Device
  HBA Port Name: /dev/cfg/c7
    Expander Device SAS Address: 500e004aaaaaaa3f
  HBA Port Name: /dev/cfg/c5
    Expander Device SAS Address: 500e004aaaaaaa3f
```

例 12 ターゲットポートを SCSI 情報とともに一覧表示する

次のコマンドは、ターゲットポートごとの SCSI 情報など、すべてのターゲットポートの詳細を一覧表示します。

例 12　ターゲットポートを SCSI 情報とともに一覧表示する　　*(続き)*

```
# sasinfo target-port -s 5000c5000bae4009
Target Port SAS Address: 5000c5000bae4009
   Type: SAS Device
        HBA Port Name: /dev/cfg/c7
            Expander Device SAS Address: 500e004aaaaaaa3f
                LUN : 0
                  OS Device Name : /dev/rdsk/c6t5000C5000BAE400Bd0s2
                  Vendor:   Sun
                  Product:  J4400
                  Device Type: Disk
        HBA Port Name: /dev/cfg/c5
            Expander Device SAS Address: 500e004aaaaaaa3f
                LUN : 0
                  OS Device Name : /dev/rdsk/c6t5000C5000BAE400Bd0s2
                  Vendor:   Sun
                  Product:  J4400
                  Device Type: Disk
```

例 13　論理ユニットを一覧表示する

次のコマンドは、ホスト上の論理ユニットを一覧表示します。

```
# sasinfo logical-unit
OS Device Name: /dev/rdsk/c4t50020F2300B4904Ed0s2
OS Device Name: /dev/rdsk/c4t50020F230000B4AFd0s2
```

例 14　論理ユニットに関する追加情報を一覧表示する

次のコマンドは、デバイス `/dev/rmt/On` に対して `-v` オプションを使用して、論理ユニットの追加情報を表示します。

```
# sasinfo lu -v
OS Device Name: /dev/rdsk/c4t50020F2300B4904Ed0s2
   HBA Port Name: /dev/cfg/c1
     Target Port SAS Address: 50020f2300b4904e
        LUN: 0
   Vendor: Sun
   Product: J4400
   Device Type: Disk
OS Device Name: /dev/rdsk/c4t50020F230000B4AFd0s2
   HBA Port Name: /dev/cfg/c1
     Target Port SAS Address: 50020f230063100b
        LUN: 0
   Vendor: Sun
   Product: J4400
   Device Type: Disk
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/sas-utilities
インタフェースの安定性	確実

関連項目

attributes(5)

名前

savecore – save a crash dump of the operating system

形式

`/usr/bin/savecore [-Lvd] [-f` *dumpfile*`] [`*directory*`]`

機能説明

The `savecore` utility saves a crash dump of the kernel (assuming that one was made) and writes a reboot message in the shutdown log. It is invoked by the `dumpadm` service each time the system boots.

`savecore` can be configured by `dumpadm(1M)` to save crash dump data in either a compressed or uncompressed format. For the compressed format, `savecore` saves the crash dump data in the file *directory*`/vmdump.`*N*, where *N* in the pathname is replaced by a number which increments by one each time `savecore` is run in *directory*. The compressed file can be uncompressed in a separate step using the `-f` *dumpfile* option. For the uncompressed format, `savecore` saves the crash dump data in the file *directory*`/vmcore.`*N* and the kernel's namelist in *directory*`/unix.`*N*.

Before writing out a crash dump, `savecore` reads a number from the file *directory*`/minfree`. This is the minimum number of kilobytes that must remain free on the file system containing *directory*. If after saving the crash dump the file system containing *directory* would have less free space the number of kilobytes specified in `minfree`, the crash dump is not saved. if the `minfree` file does not exist, `savecore` assumes a `minfree` value of 1 megabyte.

The `savecore` utility also logs a reboot message using facility `LOG_AUTH` (see `syslog(3C)`). If the system crashed as a result of a panic, `savecore` logs the panic string too.

オプション

The following options are supported:

`-d`
: Disregard dump header valid flag. Force `savecore` to attempt to save a crash dump even if the header information stored on the dump device indicates the dump has already been saved.

`-f` *dumpfile*
: Save a crash dump from the specified file instead of from the system's current dump device. When given *directory*`/vmdump.`*N*, uncompress the file to `vmcore.`*N* and `unix.`*N*, where *N* is the same number as in the compressed name.

  This option may also be useful if the information stored on the dump device has been copied to an on-disk file by means of the `dd(1M)` command.

`-L`
: Save a crash dump of the live running Solaris system, without actually rebooting or altering the system in any way. This option forces `savecore` to save a live snapshot of the system to the dump device, and then immediately to retrieve the data and to write it out to a new set of crash dump files in the specified directory. Live system crash dumps can only be performed if you have configured your system to have a dedicated dump device using `dumpadm(1M)`.

savecore -L does not suspend the system, so the contents of memory continue to change while the dump is saved. This means that live crash dumps are not fully self-consistent.

-v　Verbose. Enables verbose error messages from savecore.

オペランド

The following operands are supported:

directory　Save the crash dump files to the specified directory. If *directory* is not specified, savecore saves the crash dump files to the default savecore *directory*, configured by dumpadm(1M).

ファイル

- *directory*/vmcore.*N*
- *directory*/unix.*N*
- *directory*/bounds
- *directory*/minfree
- /var/crash/‘uname -n‘ (default crash dump directory)

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

adb(1), mdb(1), svcs(1), dd(1M), dumpadm(1M), svcadm(1M), syslog(3C), attributes(5), smf(5)

注意事項

The system crash dump service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/dumpadm:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

If the dump device is also being used as a swap device, you must run savecore very soon after booting, before the swap space containing the crash dump is overwritten by programs currently running.

名前

sbdadm – SCSI ブロックディスクコマンド行インタフェース

形式

`sbdadm create-lu [-s, --size` *size*`]` *filename*

`sbdadm delete-lu` *filename*

`sbdadm import-lu` *filename*

`sbdadm list-lu`

`sbdadm modify-lu [-s, --size` *size*`]` *lu_name* `|` *filename*

機能説明

`sbdadm` コマンドは、SCSI Target Mode Framework (STMF) に登録された、SCSI ブロックデバイスベースの論理ユニットを作成し、管理します。

サブコマンド

`sbdadm` コマンドは、次のサブコマンドをサポートします。コマンドへの引数として疑問符を入力した場合 (`sbdadm ?`)、`sbdadm` はヘルプを表示して応答します。

`create-lu [-s、--size` *size*`]` *filename*

STFM に登録できる論理ユニットを作成します。`-s` オプションでは、*size* として、整数のあとにサイズの単位を示す次のいずれかの文字を指定します。

- `k` K バイト
- `m` M バイト
- `g` G バイト
- `t` T バイト
- `p` P バイト
- `e` E バイト

size を指定しない場合、デフォルトで *filename* のサイズに設定されます。

ファイルまたはデバイスのサイズを超えるサイズを指定できます。

`delete-lu` *filename*

`sbdadm create-lu` を使用して作成された既存の論理ユニットを削除します。これにより、事実上、その論理ユニットが STMF フレームワークから読み込み解除されます。その論理ユニット上の既存のデータは、すべてそのまま維持されます。

`import-lu` *filename*

`sbdadm create-lu` を使用して以前に作成されたあとで `sbdadm delete-lu` を使用して、STMF から削除された論理ユニットを STMF にインポートおよびロードします。成功した場合、その論理ユニットは STMF から再度使用できるようになります。*filename* は、この論理ユニットに対する `sbdadm create-lu` コマンドで使用されるファイル名です。

`list-lu`

`sbdadm create-lu` コマンドを使用して作成されたすべての論理ユニットを一覧表示します。

`modify-lu` [`-s`、`--size` *size*] *lu_name* | *filename*
: `sbdadm create-lu` コマンドを使用して作成された論理ユニットの属性を変更します。`-s` オプションの場合、*size* は整数値で、数値のあとに単位指定子が続きます。単位指定子については、前述の `create-lu` の下で説明しています。このオプションが指定された場合、論理ユニットの既存のサイズは新しいサイズに変更されます。

lu_name で表されるファイルまたはデバイスのサイズを超えるサイズを指定できます。

オペランド

`sbdadm` は次のオペランドを使用します。

filename
: 既存のファイルの名前または raw ブロックデバイスへの完全指定パス。

lu_name
: 論理ユニットの 32 バイトの 16 進数表現。

使用例

例 1　論理ユニットを作成する

次の一連のコマンドは、10G バイトの論理ユニットを作成します。

```
# touch /export/lun/0
# sbdadm create-lu -s 10g /export/lun/0
# sbdadm create-lu /dev/rdsk/c1t1d0s0
```

例 2　論理ユニットを一覧表示する

次のコマンドはすべての論理ユニットを一覧表示します。

```
# sbdadm list-lu
Found 2 LU(s)

                  GUID                    DATA SIZE    SOURCE
--------------------------------  -----------  ------
6000ae4000144f21d92c47b0dd650002  10737352704  /export/lun/0
6000ae4000144f21d92c47b0de300032  134283264    /dev/rdsk/c1t1d0s0
```

終了ステータス

`0`
: 正常終了。

`0` 以外
: エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/storage/scsi-target-mode-framework
インタフェースの安定性	廃止確実

関連項目 stmfadm(1M), attributes(5)

名前

scadm – システムコントローラ (SC) の管理

形式

/usr/platform/*platform-name*/sbin/scadm *subcommand* [*option*] [*argument*]...

機能説明

scadm ユーティリティーは、システムコントローラ (SC) を管理します。このユーティリティーを使用すると、ホストサーバーが SC と対話できるようになります。

scadm ユーティリティーは、ルートとして実行しなければなりません。

scadm のインタフェース、出力、およびディレクトリ階層内の位置は確定したものではなく、変更されることがあります。

platform-name はプラットフォームの実装名です。uname -i コマンドを使用して、プラットフォーム実装を識別します。uname(1) を参照してください。

scadm ユーティリティーにはいくつかのサブコマンドがあります。一部のサブコマンドには、そのサブコマンドに関連付けられている特定のオプションと引数があります。「サブコマンド」、「オプション」、「オペランド」、および「使用法」を参照してください。

サブコマンド

コマンド行の scadm コマンドの後に SPACE を入力し、その後にサブコマンドを指定します。

次のサブコマンドがサポートされています。

consolehistory

SC のコンソールログを表示します。SC は、すべてのコンソール出力を捕捉する動的なログを維持管理します。このログは FIFO (first-in, first-out) バッファーとして管理されます。 バッファーがいっぱいになると、新しいコンソール出力が古いコンソール出力に取って代わります。デフォルトでは、コンソールログファイルの最後の 8K バイトのみが表示されます。

省略可能な -a 引数は、コンソールログファイルの全体が表示されることを表します。

consolehistory サブコマンドの読み取り速度を超えるスピードで、SC がこのログに情報を記録する可能性があります。したがって、ログデータの一部が、表示される前に失われてしまう可能性があります。そのようなことが起こった場合、consolehistory サブコマンドは、「scadm: lost <数> bytes of console log data」とログ出力内に表示し、データの一部が失われたことを通知します。

consolehistory サブコマンドの書式を次に示します。

```
scadm consolehistory [-a]
```

consolehistory サブコマンドは、すべてのプラットフォーム上で使えるわけではありません。このコマンドをサポートしないプラットフォーム上でこのコマンドが使用された場合、scadm は次のメッセージを出力します。

```
scadm: command/option not supported
```

そして、0 以外のステータスで終了します。

date
SC の日付と時刻を表示します。

date サブコマンドの書式を次に示します。

```
scadm date
```

ダウンロード
SC のファームウェアをプログラムします。

ファームウェアについては、ブートモニターとメインイメージの 2 つの部分があります。

デフォルトでは、scadm コマンドのダウンロードを実行すれば、メインファームウェアイメージがプログラムされます。boot 引数を指定すると、ブートモニターのプログラミングが選択されます。

download サブコマンドの書式を以下に示します。

```
scadm download [boot] file
```

fruhistory
SC によって維持管理される「fru (field replacable unit)」ログの内容を表示します。デフォルトでは、fru 履歴ログファイルの最後の 8K バイトのみが表示されます。このログには、SC の「showfru」コマンドのスナップショットが格納されます。このスナップショットは、システムがリセットされるか SC によってホットプラグイベントが検出されるたびに取得されます。

省略可能な -a 引数は、fru ログファイルの全体が表示されることを表します。

fruhistory サブコマンドの読み取り速度を超えるスピードで、SC がこのログに情報を記録する可能性があります。したがって、ログデータの一部が、表示される前に失われてしまう可能性があります。そのようなことが起こった場合、fruhistory サブコマンドは、「scadm: lost <数> bytes of fru log data」とログ出力内に表示し、データの一部が失われたことを通知します。

fruhistory サブコマンドの書式を次に示します。

```
scadm fruhistory [-a]
```

fruhistory サブコマンドは、すべてのプラットフォーム上で使えるわけではありません。このコマンドをサポートしないプラットフォーム上でこのコマンドが使用された場合、scadm は次のメッセージを出力します。

```
scadm: command/option not supported
```

そして、0 以外のステータスで終了します。

help
コマンドの一覧を表示します。

`help` サブコマンドの書式を以下に示します。

```
scadm help
```

loghistory
SC イベントログの最新のエントリを表示します。省略可能な `-a` 引数を指定すると、イベントログ履歴の全体が表示されます。`-a` 引数は、大規模ログファイルをサポートするプラットフォーム上でのみ使用できます。大規模ログファイルをサポートしないプラットフォーム上では、このフラグは何の効果も持ちません。

`loghistory` サブコマンドの読み取り速度を超えるスピードで、SC がこのログに情報を記録する可能性があります。したがって、ログデータの一部が、表示される前に失われてしまう可能性があります。そのようなことが起こった場合、`loghistory` サブコマンドは、「scadm: lost <数> events」とログ出力内に表示し、データの一部が失われたことを通知します。

`loghistory` サブコマンドの書式を以下に示します。

```
scadm loghistory [-a]
```

resetrsc
SC をリセットします。実行できるリセットには、ハードリセットとソフトリセットの 2 種類があります。デフォルトで実行されるのがハード リセットです。ソフトリセットを選択する場合は `-s` オプションを指定します。

`resetrsc` サブコマンドの書式を以下に示します。

```
scadm resetrsc [-s]
```

send_event
テキストベースのイベントを手動で送信します。SC は、イベントを SC イベントログに転送できます。`-c` オプションを構成して、重要な警告を電子メール、ログイン中の SC ユーザー、および `syslog` に送信できます。重要なイベントは `syslog(3C)` に記録されます。関連テキストメッセージの文字数は `80` 文字以内です。

`send_event` サブコマンドの書式を以下に示します。

```
scadm send_event [-c] "message"
```

set
SC 構成変数の値を設定します。

SC 構成変数には、次のものが含まれます。 SC IP アドレス `netsc_ipaddr` や、SC カスタマ情報 `sc_customerinfo` などです。SC 構成変数の完全なリストについては、`scadm help` コマンドの出力結果を参照してください。

`set` サブコマンドの書式を以下に示します。

```
scadm set variable value
```

show
SC 構成変数の現在の設定を表示します。変数を指定しないで scadm を実行すると、すべての変数設定が表示されます。

show サブコマンドの書式を以下に示します。

```
scadm show [variable]
```

shownetwork
SC の現在のネットワーク構成パラメータを表示します。

shownetwork サブコマンドの書式を以下に示します。

```
scadm shownetwork
```

useradd
SC にユーザーアカウントを追加します。SC では 16 名までの個別ユーザーがサポートされます。

useradd サブコマンドの書式を以下に示します。

```
scadm useradd username
```

userdel
SC からユーザーアカウントを削除します。

userdel サブコマンドの書式を以下に示します。

```
scadm userdel username
```

userpassword
指定したユーザーアカウントのパスワードを設定します。このパスワードは、現在設定されている既存のパスワードがあった場合にはそれより優先されます。新しいパスワードを設定する前に、古いパスワードは検証されません。

userpassword サブコマンドの書式を以下に示します。

```
scadm userpassword username
```

userperm
ユーザーのアクセス権のレベルを設定します。

userperm サブコマンドの書式を以下に示します。

```
scadm userperm username [aucr]
```

usershow
指定したユーザーアカウントの詳細を表示します。ユーザー名を指定しないと、すべてのユーザーアカウントの詳細が表示されます。

usershow サブコマンドの書式を次に示します。

```
scadm usershow username
```

version
SC のバージョン番号とそのコンポーネントを表示します。

version サブコマンドの書式を次に示します。

```
scadm version [-v]
```

オプション

resetrsc、send_event、および version サブコマンドには、関連するオプションがあります。コマンド行のサブコマンドの後に SPACE を入力し、その後にオプションを指定します。

resetrsc サブコマンドでは次のオプションを指定できます。

-s
ハードリセットの代わりにソフトリセットを実行します。ハードリセットは SC ハードウェアを物理的にリセットします。ソフトリセットの場合は、SC ソフトウェアがブートファームウェアにジャンプして、リセットをシミュレーションします。

send_event サブコマンドでは次のオプションを指定できます。

-c
重要なイベントを送信します。-c を指定しないで -send_event を実行すると、警告が送信されます。

version サブコマンドでは次のオプションを指定できます。

-v
バージョン番号と関連情報の詳細な内容が表示されます。

consolehistory、fruhistory、および loghistory サブコマンドは、次のオプションをサポートします。

-a
ログの全体を表示します。これらのサブコマンドは通常、最新のログデータだけを表示します。このフラグを指定すると、ログの全体が表示されるようになります。

オペランド

download, send_event, set, show, useradd, userdel, userperm, usershow, userpassword, userperm の各サブコマンドには関連する引数 (オペランド) があります。

サブコマンドにオプションを指定する場合は、コマンド行のオプションの後に SPACE を入力し、その後に引数を指定します。サブコマンドにオプションを指定しない場合は、コマンド行のサブコマンドの後に SPACE を入力し、その後に引数を指定します。引数を 2 つ以上指定する場合は、それぞれの引数の後に SPACE を入力して区切ります。

download サブコマンドでは次の引数を指定できます。

ブート
: フラッシュのブートモニター部分をプログラムします。フラッシュのメイン部分は、引数ファイルを使用せずにプログラムされます。

file
: ダウンロード用のブートまたはメインファームウェアイメージの格納場所へのパスを *file* で指定します。

  file の例は、次のとおりです。

  `/usr/platform/`*`platform_type`*`/lib/image/alommainfw`

  または

  `/usr/platform/`*`platform_type`*`/lib/image/alombootfw`

`send_event` サブコマンドでは次の引数を指定できます。

"*message*"
: *message* に指定されているテキストを使用して、イベントを記述します。*message* は引用符で囲みます。

`set` サブコマンドでは次の引数を設定できます。

variable
: SC 構成 *variable* を設定します。

value
: SC 構成変数の *value* を設定します。

`show` サブコマンドでは次の引数を指定できます。

variable
: 指定した特定の変数の値を表示します。

`useradd` サブコマンドでは次の引数を指定できます。

username
: 新しい SC アカウント *username* を追加します。

`userdel` サブコマンドでは次の引数を指定できます。

username
: SC アカウント *username* を削除します。

`userperm` サブコマンドでは次の引数を指定できます。

`-aucr`
: SC ユーザーアカウントのアクセス権を設定します。アクセス権を指定しない場合、4 つのアクセス権はすべて無効になり、読み取り専用アクセス権が割り当てられます。

  アクセス権の定義を次に示します。

a
ユーザーはSC構成変数を管理または変更できます。

u
ユーザーは、ユーザーコマンドを使用してSCアカウントを変更できます。

c
ユーザーはコンソールに接続できます。

r
ユーザーはSCのリセットと、ホストの電源投入および切断ができます。

username
SCアカウント*username*に対するアクセス権を変更します。

`-usershow`サブコマンドでは次の引数を指定できます。

username
SCアカウント*username*に関する情報を表示します。*username*を指定しないと、すべてのアカウントの情報が表示されます。

`userpassword`サブコマンドでは次の引数を指定できます。

username
*username*に対するSCパスワードを設定します。

`userperm`サブコマンドでは次の引数を指定できます。

username
*username*に対するSCアクセス権を変更します。

使用例

例1 SCの日付と時刻を表示する

次のコマンドは、SCの日付と時刻を表示します。

```
scadm date
```

例2 SCの構成変数を設定する

次のコマンドは、SCの構成変数`netsc_ipaddr`を`192.168.1.2`に設定します。

```
scadm set netsc_ipaddr 192.168.1.2
```

例3 SCの現在の構成設定を表示する

次のコマンドは、SCの現在の構成設定を表示します。

```
scadm show
```

例4 変数の現在の設定を表示する

次のコマンドは、`sys_hostname`という変数の現在の設定を表示します。

```
scadm show sys_hostname
```

例 5 テキストベースの重要なイベントを送信する

次のコマンドは、重要なイベントを SC イベントログに送信し、現在の SC ユーザーに警告するとともに、イベントを syslog(3C) に送信します。

```
scadm send_event -c "The UPS signaled a loss in power"
```

例 6 参照用のテキストベースのイベントを送信する

次のコマンドは、重要ではない参照用のテキストベースのイベントを SC イベントログに送信します。

```
scadm send_event "The disk is close to full capacity"
```

例 7 ユーザーを SC に追加する

次のコマンドは、ユーザー rscroot を SC に追加します。

```
scadm useradd rscroot
```

例 8 SC からユーザーを削除する

次のコマンドは、ユーザー olduser を SC から削除します。

```
scadm userdel olduser
```

例 9 ユーザーの詳細を表示する

次のコマンドは、すべてのユーザーアカウントの詳細を表示します。

```
scadm usershow
```

例 10 特定のユーザーの詳細を表示する

次のコマンドは、ユーザーアカウント rscroot の詳細を表示します。

```
scadm usershow rscroot
```

例 11 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー rscroot に全アクセス権 aucr を設定します。

```
scadm userperm rscroot aucr
```

例 12 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー newuser にコンソールアクセス権 c だけを設定します。

```
scadm userperm newuser c
```

例 13 ユーザーのアクセス権のレベルを設定する

次のコマンドは、ユーザー newuser のアクセス権を読み取り専用に設定します。

```
scadm userperm newuser
```

例 14　現在のネットワークパラメータを表示する

次のコマンドは、SC の現在のネットワーク構成パラメータを表示します。

```
scadm shownetwork
```

例 15　コンソール履歴を表示する

次のコマンドは、SC イベントログのコンソールの内容を表示します。

```
scadm consolehistory [-a]
```

例 16　fru 履歴を表示する

次のコマンドは、SC イベントログの「fru (field replacable unit)」の内容を表示します。

```
scadm fruhistory [-a]
```

例 17　ログ履歴を表示する

次のコマンドは、SC イベントログの最新のエントリを表示します。

```
scadm loghistory [-a]
```

例 18　詳細情報を表示する

次のコマンドは、SC とそのコンポーネントの詳細なバージョン情報を表示します。

```
scadm version -v
```

終了ステータス

次の終了値が返されます。

`0`
　正常終了。

`0` 以外
　エラーが発生しました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/library/platform
インタフェースの安定性	確実

関連項目

`uname(1)`, `syslog(3C)`, `attributes(5)`

名前

sckmd – Sun cryptographic key management daemon

形式

/usr/platform/sun4u/lib/sckmd

機能説明

sckmd is a server process that resides on a high-end system domain to maintain the Internet Protocol Security (IPsec) Security Associations (SAs) needed to secure communications between a Service Processor or System Controller (SC) and platform management software running within a domain. The cvcd(1M) and dcs(1M) daemons use these Security Associations. See ipsec(7P) for a description of Security Associations.

The sckmd daemon receives SAs from the Service Processor or SC and installs these SAs in a domain's Security Association Database (SADB) using pf_key(7P).

sckmd starts up at system boot time as an SMF service. The FMRI for the sckmd service is:

svc:/platform/sun4u/sckmd:default

A domain supports only one running sckmd process at a time.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsckmx.u, service/key-management/sun-fire-15000
Interface Stability	Committed

関連項目

cvcd(1M), dcs(1M), ipsecconf(1M), ipsecalgs(1M), attributes(5), ipsec(7P), ipsecah(7P), ipsecesp(7P), pf_key(7P)

注意事項

The sckmd service is used only on Sun Fire high-end systems and the SPARC Enterprise Server family. It provides a mechanism for exchanging IPsec keys between a domain and its System Controller (SC) or Service Processor. These platforms use IPsec to secure the communications between the SC or Service Processor and certain platform-specific daemons in the domain. Such daemons currently include cvcd(1M) and dcs(1M).

The documentation for each platform that supports sckmd describes how to configure its use of IPsec for such communications. Also, the documentation for each specific application describes how to configure its security policies and IPsec options in a manner appropriate for the target platform. Refer to the platform- and application-specific documentation for detailed information.

名前

scmadm – ストレージキャッシュマネージャーの管理ユーティリティー

形式

```
scmadm
scmadm -h
scmadm -e
scmadm -d
scmadm -v
scmadm -C [parameter [= [value]]...]
scmadm -o {system | cd | device} [option]
scmadm -m {cd | diskname | all}
```

機能説明

scmadm コマンドには、ストレージデバイスキャッシュに関する情報を制御および収集するためのさまざまなオプションが用意されています。

オプション

オプションが指定されていない場合、scmadm は、ディスク名、オプション、および汎用オプションとともに構成済みキャッシュ記述子のリストを表示します。scmadm コマンドでは次のオプションをサポートしています。

-h
: scmadm コマンドの使用法に関する情報を表示します。

-e
: 構成を読み込み、これらのパラメータでストレージデバイスキャッシュを有効にします。dscfg(1M) を参照してください。

-d
: ストレージデバイスキャッシュをシャットダウンします。

-v
: キャッシュバージョン番号を表示します。

-C [*parameter*[=[*value*]] ...]
: 構成パラメータを設定または表示します。引数を付けずに -C オプションを指定した場合、現在のキャッシュ構成パラメータが表示されます。*parameter* が指定されている場合、*parameter* の現在の値が表示されます。*parameter=value* が指定されている場合、*parameter* の現在の値が表示され、パラメータは *value* に変更されます。*value* が省略されている場合、または *value* が NULL 文字列、「」、または「-」として指定されている場合は、パラメータは構成から削除され、システムはデフォルト値を使用します。scmadm コマンドの 1 回の呼び出しで、複数のパラメータを指定できます。構成パラメータの変更は、キャッシュを次回再起動したときにはじめて有効になります。

-o { system | cd | *diskname* } [*option*]
: システムのオプションや、cd または *diskname* で指定されたキャッシュデバイスのオプションを設定または表示します。*option* rdcache または nordcache が指定されている場合、システムまたは指定されたキャッシュデバイスは、このオプ

ションに設定されます。オプションは構成の一部として保存され、持続します。dscfg(1M) を参照してください。保存したオプションを「忘れる」ようにシステムに指示するには、forget オプションを使用します。これはオプションを変更しません。保存した構成からオプションを削除するだけです。オプションをすべて省略すると、現在のオプションが表示されます。rdcache オプションがデフォルトとして設定されています。*option* は次のように定義します。

rdcache
: データブロックは、再度参照される可能性があるため、キャッシュに残すようにします。

nordcache
: データブロックは再度参照される可能性はなく、使用頻度がもっとも低いものとして扱うようにします。その結果、ほかのブロックをキャッシュに長い間残すことができます。

-m { cd | *diskname* | all }
: cd または *diskname* で指定されたデバイスのキャッシュ記述子とディスク名のマップを表示します。または、all を指定した場合は、システム上のすべてのストレージデバイスのキャッシュ記述子とディスク名のマップを表示します。

終了ステータス

scmadm コマンドは成功した場合に 0 を、エラーの場合に 0 以外を返します。

ファイル

/dev/sdbc

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-cache-management
インタフェースの安定性	確実

関連項目

dscfg(1M), attributes(5)

診断

連続したメモリーが不十分な場合、scmadm は失敗します。

名前

sconadm – register system information

形式

```
/usr/sbin/sconadm register -a
     [-e softwareUpdate | -E softwareUpdate]
     [-h hostname] [-l logfile] [-N]
     [-p proxy_host[:proxy_port]]
     [-r registration_profile] [-u username]
     [-x proxy_username]

/usr/sbin/sconadm proxy [-l logfile]
     [-p proxy_host[:proxy_port]]
     [-r registration_profile] [-x proxy_username]
```

機能説明

The `sconadm` utility is a command-line version of the Basic Registration GUI. In the first form of the command in the SYNOPSIS, `sconadm` uses the `register` subcommand to register a host with a registration server. In the second form, `sconadm` uses the `proxy` subcommand to configure all of the components for software update to use an HTTP web proxy.

The parameters specified with `-u`, `-e` (or `-E`), `-h`, `-p`, and `-x` override values specified in your registration profile. A template for this profile, owned by root, with read-only permissions, is stored in `/usr/lib/breg/data/RegistrationProfile.properties`. See `registration_profile(4)`.

For the `proxy` subcommand, the proxy password is stored in the `RegistrationProfile.properties` file, available if proxy authentication is needed. Storage in the profile prevents proxy passwords from being exposed as part of a listing of processes on a system.

オプション

The following options are supported:

`-a`	Accept “Terms of Use and Binary Code License”. Absence of this option means that you do not accept the license.
`-e` softwareUpdate	Enable client to be managed at the Sun-hosted Update Connection Service.
`-E` softwareUpdate	Disable client's ability to be managed at the Sun-hosted Update Connection Service.
`-h` *hostname*	Hostname of the machine you want to register.
`-l` *logfile*	Pathname of log file.
`-N`	Never register.
`-p` *proxy_host*[:*proxy_port*]	Proxy host name and optional proxy port number.
`-r` *registration_profile*	Pathname to a registration profile.
`-u` *username*	User name (a Sun Online Account).
`-x` *proxy_username*	User name on the proxy host.

使用例

Unless specified otherwise, the commands below require root privileges or privileges equivalent to root. See `privileges(5)`.

例 1 Registering a New System

Assume a file `registrationprofile.properties` in `/tmp` that contains the following:

```
userName=user123
password=abc123
hostName=
subscriptionKey=
portalEnabled=false
proxyHostName=
proxyPort=
proxyUserName=
proxyPassword=
```

To register a new system using the profile above, you enter:

```
/usr/sbin/sconadm register -a -r /tmp/registrationprofile.properties
```

例 2 Reregistering a System with a Different User

Assume a file `registrationprofile.properties` in `/tmp` with the contents shown below. Note the changed specification for `userName` and `password`.

```
userName=newuser
password=newpassword
hostName=
subscriptionKey=
portalEnabled=false
proxyHostName=
proxyPort=
proxyUserName=
proxyPassword=
```

To reregister a new system using the profile above, you enter the same command you entered to register the system:

```
/usr/sbin/sconadm register -a -r /tmp/registrationprofile.properties
```

例 3 Reregistering a System, Adding a Sun Subscription Key

Modify `registrationprofile.properties` as follows:

```
userName=newuser
password=newpassword
hostName=
subscriptionKey=abc12345678
portalEnabled=false
proxyHostName=
```

例 3 Reregistering a System, Adding a Sun Subscription Key *(続き)*

```
proxyPort=
proxyUserName=
proxyPassword=
```

Run the command:

```
/usr/sbin/sconadm register -a -r /tmp/registrationprofile.properties
```

例 4 Reregistering and Enabling Access to all Update Connection Services

Modify `registrationprofile.properties` as follows:

```
userName=newuser
password=newpassword
hostName=
subscriptionKey=abc12345678
portalEnabled=false
proxyHostName=
proxyPort=
proxyUserName=
proxyPassword=
```

Note that `portalEnabled` is set to `false`. Run the command:

```
/usr/sbin/sconadm register -a -r /tmp/registrationprofile.properties \
-e softwareUpdate
```

例 5 Never Registering

To never register a system, enter:

```
/usr/sbin/sconadm register -N
```

例 6 Using a Proxy Server With Proxy Authentication

Edit `registrationprofile.properties` as follows:

```
userName=
password=
hostName=
subscriptionKey=
portalEnabled=
proxyHostName=webcache.mycompany.com
proxyPort=8080
proxyUserName=myCompanyProxyUserName
proxyPassword=myCompanyProxyPassword
```

Run the command:

例 6 Using a Proxy Server With Proxy Authentication *(続き)*

```
/usr/sbin/sconadm proxy -r /tmp/registrationprofile.properties
```

例 7 Changing Proxy Host Settings

Edit `registrationprofile.properties` as follows:

```
userName=
password=
hostName=
subscriptionKey=
portalEnabled=
proxyHostName=webcache.mycompany.com
proxyPort=8080
proxyUserName=myCompanyProxyUserName
proxyPassword=myCompanyProxyPassword
```

Run the command:

```
/usr/sbin/sconadm proxy -r /tmp/registrationprofile.properties
```

Then, change the `proxyHostName` value by running the following command:

```
/usr/sbin/sconadm proxy -r /tmp/registrationprofile.properties \
-p newproxy.mycompany.com
```

After the preceding command all proxies use `newproxy.mycompany.com`.

例 8 Resetting a System Not to Use a Proxy

Edit `registrationprofile.properties` as follows:

```
userName=
password=
hostName=
subscriptionKey=
portalEnabled=
proxyHostName=
proxyPort=
proxyUserName=
proxyPassword=
```

Note that values for all proxy fields are null.

Run the command:

```
/usr/sbin/sconadm proxy -r /tmp/registrationprofile.properties
```

終了ステータス

`0` Success.

`>0` An error occurred.

ファイル

`/usr/lib/breg/data/RegistrationProfile.properties`
Registration profile template.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWbrg
Interface Stability	Committed

関連項目

`registration_profile(4)`, `attributes(5)`, `privileges(5)`

名前

sdpadm – SDP system configuration administration

形式

```
/usr/sbin/sdpadm status | enable | disable
```

機能説明

The `sdpadm` command is no longer shipped with the Oracle Solaris operating system. SDP sockets are enabled by default. This man page describes a utility that was shipped in prior builds of the operating system.

The `sdpadm` command is used to display the system state of the Sockets Direct Protocol (SDP) protocol. The `sdpadm` command can optionally be used to set the state of the SDP protocol. See `sdp(7D)`.

By default the SDP protocol is disabled on the system. It can be enabled by using `sdpadm enable`.

オプション

The following subcommands are supported:

`status` Displays the system status of the SDP protocol

`enable` Enables the SDP protocol

`disable` Disables the SDP protocol

使用法

The required privileges to change the state of the SDP protocol are controlled by the network configuration policy. If a user does not have the correct privileges to set the SDP policy, `sdpadm` returns the current state of SDP without having changed the state.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/spdadm
Interface Stability	Obsolete

関連項目

`attributes(5)`, `sdp(7D)`

Infiniband Specification Volume 1 (`http://www.infinibandta.org`)

名前

sendmail – send mail over the internet

形式

```
/usr/lib/sendmail [-Ac] [-Am] [-ba] [-bD] [-bd] [-bi] [-bl]
     [-bm] [-bp] [-bP] [-bs] [-bt] [-bv] [-B type] [-C file]
     [-D logfile] [-d X] [-F fullname] [-f name] [-G]
     [-h N] [-L tag] [-M xvalue] [-N notifications] [-n]
     [-Ooption =value] [-o xvalue] [-p protocol]
     [-Q [reason]] [-q [time]] [-q Xstring] [-R ret]
     [-r name] [-t] [-V envid] [-v] [-X logfile]
     [address]...
```

機能説明

The `sendmail` utility sends a message to one or more people, routing the message over whatever networks are necessary. `sendmail` does internetwork forwarding as necessary to deliver the message to the correct place.

`sendmail` is not intended as a user interface routine. Other programs provide user-friendly front ends. `sendmail` is used only to deliver pre-formatted messages.

With no flags, `sendmail` reads its standard input up to an EOF, or a line with a single dot, and sends a copy of the letter found there to all of the addresses listed. It determines the network to use based on the syntax and contents of the addresses.

Local addresses are looked up in the local `aliases(4)` file, or in a name service as defined by the `nsswitch.conf(4)` file, and aliased appropriately. In addition, if there is a `.forward` file in a recipient's home directory, `sendmail` forwards a copy of each message to the list of recipients that file contains. Refer to the `NOTES` section for more information about `.forward` files. Aliasing can be prevented by preceding the address with a backslash.

There are several conditions under which the expected behavior is for the alias database to be either built or rebuilt. This cannot occur under any circumstances unless root owns *and* has exclusive write permission to the `/etc/mail/aliases*` files.

If a message is found to be undeliverable, it is returned to the sender with diagnostics that indicate the location and nature of the failure; or, the message is placed in a `dead.letter` file in the sender's home directory.

Service Management

The `sendmail` service is managed by the service management facility, `smf(5)`, under the service identifiers:

```
svc:/network/smtp:sendmail
svc:/network/sendmail-client:default
```

Administrative actions on these services, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The services' status can be queried using the `svcs(1)` command.

These are separate services rather than instances of the same service so that other services can properly express any dependencies. In particular, here are some guidelines about which service/instance should be depended on for which purposes:

- For a service that uses sendmail to send mail, an optional dependency on the service svc:/network/sendmail-client might be in order.
- For a service that needs to receive mail in general, but does not depend on sendmail being the particular SMTP receiver, a dependency on the service svc:/network/smtp might be in order.
- For a service that needs to interact with sendmail in particular, such as a Milter, a dependency on the instance svc:/network/smtp:sendmail might be in order.

For the last two, note the difference, as the latter has the ":sendmail" instance specification, whereas the former does not, thus representing the more general service.

Enabling Access to Remote Clients

On an unmodified system, access to sendmail by remote clients is enabled and disabled through the service management facility (see smf(5)). In particular, remote access is determined by the value of the local_only SMF property:

```
svc:/network/smtp:sendmail/config/local_only = true
```

A setting of true, as above, disallows remote access; false allows remote access. The default value is true.

The following example shows the sequence of SMF commands used to enable sendmail to allow access to remote systems:

```
# svccfg -s svc:/network/smtp:sendmail setprop config/local_only = false
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

See svcadm(1M) and svccfg(1M).

Note, however, on a system where any of the sendmail(4) files have been customized, setting this property might not have the intended effect. See sendmail(4) for details.

Automated Rebuilding of Configuration Files

See sendmail(4) for details on which service properties can be set to automate (re)building of configuration files when the service is started.

Restricting Host Access

sendmail uses TCP Wrappers to restrict access to hosts. It uses the service name of sendmail for hosts_access(). For more information on TCP Wrappers, see tcpd(1M) and hosts_access(4) in the security/tcp-wrapper package. tcpd(1M) and hosts_access(4) are not part of the Solaris man pages.

Startup Options

The /etc/default/sendmail file stores startup options for sendmail so that the options are not removed when a host is upgraded. See also sendmail(4) for details on which service properties can be set to automate (re)building of configuration files when the service is started.

You can use the following variables in the /etc/default/sendmail startup file:

CLIENTOPTIONS=*string*
: Selects additional options to be used with the client daemon, which looks in the client-only queue (`/var/spool/clientmqueue`) and acts as a client queue runner. No syntax checking is done, so be careful when making changes to this variable.

CLIENTQUEUEINTERVAL=#
: Similar to the `QUEUEINTERVAL` option, `CLIENTQUEUEINTERVAL` sets the time interval for mail queue runs. However, the `CLIENTQUEUEINTERVAL` option controls the functions of the client daemon, instead of the functions of the master daemon. Typically, the master daemon is able to deliver all messages to the SMTP port. However, if the message load is too high or the master daemon is not running, then messages go into the client-only queue, `/var/spool/clientmqueue`. The client daemon, which checks in the client-only queue, then acts as a client queue processor.

ETRN_HOSTS=*string*
: Enables an SMTP client and server to interact immediately without waiting for the queue run intervals, which are periodic. The server can immediately deliver the portion of its queue that goes to the specified hosts. For more information, refer to the `etrn(1M)` man page.

MODE=-bd
: Selects the mode to start `sendmail` with. Use the `-bd` option or leave it undefined.

OPTIONS=*string*
: Selects additional options to be used with the master daemon. No syntax checking is done, so be careful when making changes to this variable.

QUEUEINTERVAL=#
: Sets the interval for mail queue runs on the master daemon. # can be a positive integer that is followed by either `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks. The syntax is checked before `sendmail` is started. If the interval is negative or if the entry does not end with an appropriate letter, the interval is ignored and `sendmail` starts with a queue interval of 15 minutes.

QUEUEOPTIONS=p
: Enables one persistent queue runner that sleeps between queue run intervals, instead of a new queue runner for each queue run interval. You can set this option to `p`, which is the only setting available. Otherwise, this option is not set.

Mail Filter API

`sendmail` supports a mail filter API called “milter”. For more information, see `/usr/include/libmilter/README` and http://www.milter.org

オプション

The following options are supported:

`-Ac`
: Uses `submit.cf` even if the operation mode does not indicate an initial mail submission.

`-Am`
: Uses `sendmail.cf` even if the operation mode indicates an initial mail submission.

`-ba`
Goes into `ARPANET` mode. All input lines must end with a RETURN-LINEFEED, and all messages are generated with a RETURN-LINEFEED at the end. Also, the `From:` and `Sender:` fields are examined for the name of the sender.

`-bd`
Runs as a daemon in the background, waiting for incoming SMTP connections.

`-bD`
Runs as a daemon in the foreground, waiting for incoming SMTP connections.

`-bi`
Initializes the `aliases(4)` database. Root must own *and* have exclusive write permission to the `/etc/mail/aliases*` files for successful use of this option.

`-bl`
Runs as a daemon (like `-bd`) but accepts only loopback SMTP connections.

`-bm`
Delivers mail in the usual way (default).

`-bp`
Prints a summary of the mail queues.

`-bP`
Prints the number of entries in the queues. This option is only available with shared memory support.

`-bs`
Uses the SMTP protocol as described in RFC 2821. This flag implies all the operations of the `-ba` flag that are compatible with SMTP.

`-bt`
Runs in address test mode. This mode reads addresses and shows the steps in parsing; it is used for debugging configuration tables.

`-bv`
Verifies names only. Does not try to collect or deliver a message. Verify mode is normally used for validating users or mailing lists.

`-B` *type*
Indicates body *type* (`7BIT` or `8BITMIME`).

`-C` *file*
Uses alternate configuration file.

`-D` *logfile*
Send debugging output to the indicated log file instead of `stdout`.

`-d` *X*
Sets debugging value to *X*.

`-f` *name*
Sets the name of the "from" person (that is, the sender of the mail).

`-F` *fullname*
Sets the full name of the sender.

`-G`
When accepting messages by way of the command line, indicates that they are for relay (gateway) submission. When this flag is set, `sendmail` might complain about syntactically invalid messages, for example, unqualified host names, rather than fixing them. `sendmail` does not do any canonicalization in this mode.

`-h` *N*
Sets the hop count to *N*. The hop count is incremented every time the mail is processed. When it reaches a limit, the mail is returned with an error message, the victim of an aliasing loop.

`-L` *tag*
Sets the identifier used in `syslog` messages to the supplied *tag*.

`-M`*xvalue*
Sets macro *x* to the specified *value*.

`-n`
Does not do aliasing.

`-N` *notifications*
Tags all addresses being sent as wanting the indicated *notifications*, which consists of the word "NEVER" or a comma-separated list of "SUCCESS", "FAILURE", and "DELAY" for successful delivery, failure and a message that is stuck in a queue somwhere. The default is "FAILURE,DELAY".

`-o`*xvalue*
Sets option *x* to the specified *value*. `Processing Options` are described below.

`-O`*option=value*
Sets *option* to the specified *value* (for long from names). `Processing Options` are described below.

`-p` *protocol*
Sets the sending protocol. The *protocol* field can be in form *protocol*:*host* to set both the sending protocol and the sending host. For example: `-pUUCP:uunet` sets the sending *protocol* to `UUCP` and the sending host to `uunet`. Some existing programs use `-oM` to set the *r* and *s* macros; this is equivalent to using `-p`.

`-q`[*time*]
Processes saved messages in the queue at given intervals. If *time* is omitted, processes the queue once. *time* is given as a tagged number, where *s* is seconds, *m* is minutes, *h* is hours, *d* is days, and *w* is weeks. For example, `-q1h30m` or `-q90m` would both set the timeout to one hour thirty minutes.

By default, sendmail runs in the background. This option can be used safely with `-bd`.

`-qp[`*time*`-]`
Similar to `-q[`*time*`]`, except that instead of periodically forking a child to process the queue, `sendmail` forks a single persistent child for each queue that alternates between processing the queue and sleeping. The sleep time (*time*) is specified as the argument; it defaults to `1` second. The process always sleeps at least `5` seconds if the queue was empty in the previous queue run.

`-qf`
Processes saved messages in the queue once and does not `fork(2)`, but runs in the foreground.

`-qG` *name*
Processes jobs in queue group called *name* only.

`-q[!]I` *substr*
Limits processed jobs to those containing *substr* as a substring of the queue ID or not when `!` is specified.

`-q[!]Q` *substr*
Limits processed jobs to those quarantined jobs containing *substr* as a substring of the quarantine *reason* or not when `!` is specified.

`-q[!]R` *substr*
Limits processed jobs to those containing *substr* as a substring of one of the recipients or not when `!` is specified.

`-q[!]S` *substr*
Limits processed jobs to those containing *substr* as a substring of the sender or not when `!` is specified.

`-Q[`*reason*`]`
Quarantines a normal queue item with the given reason or unquarantines a quarantined queue item if no reason is given. This should only be used with some sort of item matching as described above.

`-r` *name*
An alternate and obsolete form of the `-f` flag.

`-R` *ret*
Identifies the information you want returned if the message bounces. *ret* can be `HDRS` for headers only or `FULL` for headers plus body.

`-t`
Reads message for recipients. `To:`,`Cc:`, and `Bcc:` lines are scanned for people to send to. The `Bcc:` line is deleted before transmission. Any addresses in the argument list is suppressed. The `NoRecipientAction` Processing Option can be used to change the behavior when no legal recipients are included in the message.

`-v`
Goes into verbose mode. Alias expansions are announced, and so forth.

`-V` *envid*
The indicated *envid* is passed with the envelope of the message and returned if the message bounces.

`-X` *logfile*
Logs all traffic in and out of `sendmail` in the indicated *logfile* for debugging mailer problems. This produces a lot of data very quickly and should be used sparingly.

Processing Options

There are a number of "random" options that can be set from a configuration file. Options are represented by a single character or by multiple character names. The syntax for the single character names of is:

`O`*xvalue*

This sets option *x* to be *value*. Depending on the option, *value* may be a string, an integer, a boolean (with legal values `t`, `T`, `f`, or `F`; the default is `TRUE`), or a time interval.

The multiple character or long names use this syntax:

`O` *Longname=argument*

This sets the option *Longname* to be *argument*. The long names are beneficial because they are easier to interpret than the single character names.

Not all processing options have single character names associated with them. In the list below, the multiple character name is presented first followed by the single character syntax enclosed in parentheses.

`AliasFile (A`*file*`)`
Specifies possible alias files.

`AliasWait (a` *N*`)`
If set, waits up to *N* minutes for an "@:@" entry to exist in the `aliases(4)` database before starting up. If it does not appear in *N* minutes, issues a warning. Defaults to 10 minutes.

`AllowBogusHELO`
Allows a `HELO SMTP` command that does not include a host name. By default this option is disabled.

`BadRcptThrottle=`*N*
If set and more than the specified number of recipients in a single SMTP envelope are rejected, sleeps for one second after each rejected RCPT command.

`BlankSub (B`*c*`)`
Sets the blank substitution character to *c*. Unquoted spaces in addresses are replaced by this character. Defaults to SPACE (that is, no change is made).

`CACertFile`
File containing one CA cert.

`CACertPath`
Path to directory with certs of CAs.

`CheckAliases (n)`
Validates the RHS of aliases when rebuilding the `aliases(4)` database.

`CheckpointInterval (C`*N*`)`
Checkpoints the queue every *N* (default `10`) addresses sent. If your system crashes during delivery to a large list, this prevents retransmission to any but the last *N* recipients.

`ClassFactor (z`*fact*`)`
The indicated factor *fact* is multiplied by the message class (determined by the `Precedence:` field in the user header and the `P` lines in the configuration file) and subtracted from the priority. Thus, messages with a higher `Priority:` are favored. Defaults to `1800`.

`ClientCertFile`
File containing the cert of the client, that is, this cert is used when `sendmail` acts as client.

`ClientKeyFile`
File containing the private key belonging to the client cert.

`ClientPortOptions`
Sets client SMTP options. The options are key=value pairs. Known keys are:

`Addr` *Address Mask*
Address Mask defaults to `INADDR_ANY`. The address mask can be a numeric address in dot notation or a network name.

`Family`
Address family (defaults to INET).

`Listen`
Size of listen queue (defaults to 10).

`Port`
Name/number of listening port (defaults to `smtp`).

`RcvBufSize`
The size of the TCP/IP receive buffer.

`SndBufSize`
The size of the TCP/IP send buffer.

`Modifier`
Options (flags) for the daemon. Can be:

`h`
Uses name of interface for `HELO` command.

If h is set, the name corresponding to the outgoing interface address (whether chosen by means of the Connection parameter or the default) is used for the HELO/EHLO command.

ColonOkInAddr
If set, colons are treated as a regular character in addresses. If not set, they are treated as the introducer to the RFC 822 "group" syntax. This option is on for version 5 and lower configuration files.

ConnectionCacheSize (k*N*)
The maximum number of open connections that are to be cached at a time. The default is 1. This delays closing the current connection until either this invocation of sendmail needs to connect to another host or it terminates. Setting it to 0 defaults to the old behavior, that is, connections are closed immediately.

ConnectionCacheTimeout (K*timeout*)
The maximum amount of time a cached connection is permitted to idle without activity. If this time is exceeded, the connection is immediately closed. This value should be small (on the order of ten minutes). Before sendmail uses a cached connection, it always sends a NOOP (no operation) command to check the connection. If the NOOP command fails, it reopens the connection. This keeps your end from failing if the other end times out. The point of this option is to be a good network neighbor and avoid using up excessive resources on the other end. The default is five minutes.

ConnectionRateThrottle
The maximum number of connections permitted per second. After this many connections are accepted, further connections are delayed. If not set or <= 0, there is no limit.

ConnectionRateWindowSize
Define the length of the interval for which the number of incoming connections is maintained. The default is 60 seconds.

ControlSocketName
Name of the control socket for daemon management. A running sendmail daemon can be controlled through this Unix domain socket. Available commands are: help, restart, shutdown, and status. The status command returns the current number of daemon children, the free disk space (in blocks) of the queue directory, and the load average of the machine expressed as an integer. If not set, no control socket is available. For the sake of security, this Unix domain socket must be in a directory which is accessible only by root; /var/spool/mqueue/.smcontrol is recommended for the socket name.

CRLFile
File containing certificate revocation status, useful for X.509v3 authentication.

DaemonPortOptions (O*options*)
Sets server SMTP options. The options are *key*=*value* pairs. Known keys are:

Name
User-definable name for the daemon (defaults to "Daemon#"). Used for error messages and logging.

`Addr`
: Address mask (defaults `INADDR_ANY`).

  The address mask may be a numeric address in dot notation or a network name.

`Family`
: Address family (defaults to `INET`).

`InputMailFilters`
: List of input mail filters for the daemon.

`Listen`
: Size of listen queue (defaults to `10`).

`Modifier`
: Options (flags) for the daemon; can be a sequence (without any delimiters) of:

  `a`
  : Requires authentication.

  `b`
  : Binds to interface through which mail has been received.

  `c`
  : Performs hostname canonification (`.cf`).

  `f`
  : Requires fully qualified hostname (`.cf`).

  `h`
  : Uses name of interface for `HELO` command.

  `u`
  : Allows unqualified addresses (`.cf`).

  `C`
  : Does not perform hostname canonification.

  `E`
  : Disallows `ETRN` (see RFC 2476).

`Name`
: User-definable name for the daemon (defaults to `Daemon#`). Used for error messages and logging.

`Port`
: Name/number of listening port (defaults to `smtp`).

`ReceiveSize`
: The size of the TCP/IP receive buffer.

`SendSize`
: The size of the TCP/IP send buffer.

`children`
Maximum number of children per daemon. See `MaxDaemonChildren`.

`DeliveryMode`
Delivery mode per daemon. See `DeliveryMode`.

`refuseLA`
`RefuseLA` per daemon.

`delayLA`
`DelayLA` per daemon.

`queueLA`
`QueueLA` per daemon.

`sendmail` listens on a new socket for each occurrence of the `DaemonPortOptions` option in a configuration file.

`DataFileBufferSize`
Sets the threshold, in bytes, before a memory-bases queue data file becomes disk-based. The default is 4096 bytes.

`DeadLetterDrop`
Defines the location of the system-wide dead.letter file, formerly hard-coded to `/var/tmp/dead.letter`. If this option is not set (the default), `sendmail` does not attempt to save to a system-wide `dead.letter` file in the event it cannot bounce the mail to the user or postmaster. Instead, it renames the `qf` file as it has in the past when the `dead.letter` file could not be opened.

`DefaultCharSet`
Sets the default character set to use when converting unlabeled 8 bit input to MIME.

`DefaultUser (g`*gid*`) or (u`*uid*`)`
Sets the default group ID for mailers to run in to *gid* or set the default userid for mailers to *uid*. Defaults to `1`. The value can also be given as a symbolic group or user name.

`DelayLA=`*LA*
When the system load average exceeds *LA*, `sendmail` sleeps for one second on most SMTP commands and before accepting connections.

`DeliverByMin=`*time*
Sets minimum time for `Deliver By SMTP Service Extension` (RFC 2852). If `0`, no time is listed, if less than `0`, the extension is not offered, if greater than `0`, it is listed as minimum time for the `EHLO` keyword `DELIVERBY`.

`DeliveryMode (d`*x*`)`
Delivers in mode *x*. Legal modes are:

`i`
Delivers interactively (synchronously).

`b`
Delivers in background (asynchronously).

`d`
Deferred mode. Database lookups are deferred until the actual queue run.

`q`
Just queues the message (delivers during queue run).

Defaults to `b` if no option is specified, `i` if it is specified but given no argument (that is, `Od` is equivalent to `Odi`).

`DHParameters`
File containing the DH parameters.

`DialDelay`
If a connection fails, waits this many seconds and tries again. Zero means "do not retry".

`DontBlameSendmail`
If set, overrides the file safety checks. This compromises system security and should not be used. See http://www.sendmail.org/tips/dontBlameSendmail for more information.

`DontExpandCnames`
If set, $[... $] lookups that do DNS-based lookups do not expand CNAME records.

`DontInitGroups`
If set, the `initgroups(3C)` routine is never invoked. If you set this, agents run on behalf of users only have their primary (`/etc/passwd`) group permissions.

`DontProbeInterfaces`
If set, `sendmail` does not insert the names and addresses of any local interfaces into the `$=w class`. If set, you must also include support for these addresses, otherwise mail to addresses in this list bounces with a configuration error.

`DontPruneRoutes (R)`
If set, does not prune route-addr syntax addresses to the minimum possible.

`DoubleBounceAddress`
If an error occurs when sending an error message, sends that "double bounce" error message to this address.

`EightBitMode (8)`
Uses 8–bit data handling. This option requires one of the following keys. The key can selected by using just the first character, but using the full word is better for clarity.

`mimify`
Does any necessary conversion of `8BITMIME` to 7–bit.

`pass`
Passes unlabeled 8–bit input through as is.

`strict`
Rejects unlabeled 8–bit input.

`ErrorHeader (E`*file/message*`)`
Appends error messages with the indicated message. If it begins with a slash, it is assumed to be the pathname of a file containing a message (this is the recommended setting). Otherwise, it is a literal message. The error file might contain the name, email address, and/or phone number of a local postmaster who could provide assistance to end users. If the option is missing or `NULL`, or if it names a file which does not exist or which is not readable, no message is printed.

`ErrorMode (e`*x*`)`
Disposes of errors using mode *x*. The values for *x* are:

`e`
Mails back errors and gives `0` exit status always.

`m`
Mails back errors.

`p`
Prints error messages (default).

`q`
No messages, just gives exit status.

`w`
Writes back errors (mail if user not logged in).

`FallbackMXhost (V`*fallbackhost*`)`
If specified, the *fallbackhost* acts like a very low priority MX on every host. This is intended to be used by sites with poor network connectivity.

`FallBackSmartHost`
If specified, the *fallBackSmartHost* is used in a last-ditch effort for each host. This is intended to be used by sites with “fake internal DNS”. That is, a company whose DNS accurately reflects the world inside that company's domain but not outside.

`FastSplit`
If set to a value greater than zero (the default is one), it suppresses the MX lookups on addresses when they are initially sorted, that is, for the first delivery attempt. This usually results in faster envelope splitting unless the MX records are readily available in a local DNS cache. To enforce initial sorting based on MX records set `FastSplit` to zero. If the mail is submitted directly from the command line, then the value also limits the number of processes to deliver the envelopes; if more envelopes are created they are only queued up and must be taken care of by a queue run. Since the default submission method is by way of SMTP (either from a MUA or by way of the Message Submission Program [MSP]), the value of `FastSplit` is seldom used to limit the number of processes to deliver the envelopes.

`ForkEachJob (Y)`
: If set, delivers each job that is run from the queue in a separate process. Use this option if you are short of memory, since the default tends to consume considerable amounts of memory while the queue is being processed.

`ForwardPath (J`*path*`)`
: Sets the path for searching for users' `.forward` files. The default is `$z/.forward`. Some sites that use the automounter may prefer to change this to `/var/forward/$u` to search a file with the same name as the user in a system directory. It can also be set to a sequence of paths separated by colons; `sendmail` stops at the first file it can successfully and safely open. For example, `/var/forward/$u:$z/.forward` searches first in `/var/forward/` *username* and then in ~*username*`/.forward` (but only if the first file does not exist). Refer to the NOTES section for more information.

`HeloName=`*name*
: Sets the name to be used for `HELO`/`EHLO` (instead of `$j`).

`HelpFile (H`*file*`)`
: Specifies the help file for SMTP.

`HoldExpensive (c)`
: If an outgoing mailer is marked as being expensive, does not connect immediately.

`HostsFile`
: Sets the file to use when doing "file" type access of host names.

`HostStatusDirectory`
: If set, host status is kept on disk between `sendmail` runs in the named directory tree. If a full path is not used, then the path is interpreted relative to the queue directory.

`IgnoreDots (i)`
: Ignores dots in incoming messages. This is always disabled (that is, dots are always accepted) when reading SMTP mail.

`LogLevel (L`*n*`)`
: Sets the default log level to *n*. Defaults to `9`.

`(M`*x value*`)`
: Sets the macro *x* to *value*. This is intended only for use from the command line.

`MailboxDatabase`
: Type of lookup to find information about local mail boxes, defaults to `pw` which uses `getpwnam(3C)`. Other types can be introduced by adding them to the source code, see `libsm/mbdb.c` for details.

`MatchGECOS (G)`
: Tries to match recipient names using the GECOS field. This allows for mail to be delivered using names defined in the GECOS field in `/etc/passwd` as well as the login name.

`MaxDaemonChildren`
: The maximum number of children the daemon permits. After this number, connections are rejected. If not set or <=0, there is no limit.

`MaxHopCount (h`*N*`)`
: The maximum hop count. Messages that have been processed more than *N* times are assumed to be in a loop and are rejected. Defaults to `25`.

`MaxMessageSize`
: The maximum size of messages that are accepted (in bytes).

`MaxMimeHeaderLength=`*M*`[/`*N*`]`
: Sets the maximum length of certain MIME header field values to *M* characters. For some of these headers which take parameters, the maximum length of each parameter is set to *N* if specified. If `/`*N* is not specified, one half of *M* is used. By default, these values are `0`, meaning no checks are done.

`MaxNOOPCommands=`*N*
: Overrides the default of 20 for the number of useless commands.

`MaxQueueChildren=`*N*
: When set, this limits the number of concurrent queue runner processes to *N*. This helps to control the amount of system resources used when processing the queue. When there are multiple queue groups defined and the total number of queue runners for these queue groups would exceed `MaxQueueChildren` then the queue groups are not all run concurrently. That is, some portion of the queue groups run concurrently such that `MaxQueueChildren` is not be exceeded, while the remaining queue groups are run later (in round robin order). See `MaxRunnersPerQueue`.

`MaxQueueRunSize`
: If set, limits the maximum size of any given queue run to this number of entries. This stops reading the queue directory after this number of entries is reached; job priority is not used. If not set, there is no limit.

`MaxRunnersPerQueue=`*N*
: This sets the default maximum number of queue runners for queue groups. Up to *N* queue runners work in parallel on a queue group's messages. This is useful where the processing of a message in the queue might delay the processing of subsequent messages. Such a delay can be the result of non-erroneous situations such as a low bandwidth connection. The can be overridden on a per queue group basis by setting the `Runners` option. The default is `1` when not set.

`MeToo (m)`
: Sends to me too, even if I am in an alias expansion.

`MaxRecipientsPerMessage`
: If set, allows no more than the specified number of recipients in an SMTP envelope. Further recipients receive a 452 error code and are deferred for the next delivery attempt.

`MinFreeBlocks (b`*N*`/`*M*`)`
: Insists on at least *N* blocks free on the file system that holds the queue files before accepting email by way of SMTP. If there is insufficient space, `sendmail` gives a `452` response to the `MAIL` command. This invites the sender to try again later. The optional *M* is a maximum message size advertised in the `ESMTP EHLO` response. It is currently otherwise unused.

`MinQueueAge`
: Specifies the amount of time a job must sit in the queue between queue runs. This allows you to set the queue run interval low for better responsiveness without trying all jobs in each run. The default value is 0.

`MustQuoteChars`
: Specifies the characters to be quoted in a full name phrase. `&,;:\()[]` are quoted automatically.

`NiceQueueRun`
: Specifies the priority of queue runners. See `nice(1)`.

`NoRecipientAction`
: Sets action if there are no legal recipient files in the message. The legal values are:

  `add-apparently-to`
  : Adds an `Apparently-to:` header with all the known recipients (which may expose blind recipients).

  `add-bcc`
  : Adds an empty `Bcc:` header.

  `add-to`
  : Adds a `To:` header with all the known recipients (which may expose blind recipients).

  `add-to-undisclosed`
  : Adds a `To: undisclosed-recipients:` header.

  `none`
  : Does nothing, that is, leaves the message as it is.

`OldStyleHeaders (o)`
: Assumes that the headers may be in old format, that is, spaces delimit names. This actually turns on an adaptive algorithm: if any recipient address contains a comma, parenthesis, or angle bracket, it is assumed that commas already exist. If this flag is not on, only commas delimit names. Headers are always output with commas between the names.

`OperatorChars or $o`
: Defines the list of characters that can be used to separate the components of an address into tokens.

`PidFile`
: Specifies the filename of the `pid` file. The default is `/var/run/sendmail.pid`. The filename is macro-expanded before it is opened, and unlinked when `sendmail` exits.

`PostmasterCopy (P`*postmaster*`)`
: If set, copies of error messages are sent to the named *postmaster*. Only the header of the failed message is sent. Since most errors are user problems, this is probably not a good idea on large sites, and arguably contains all sorts of privacy violations, but it seems to be popular with certain operating systems vendors.

`PrivacyOptions (p`*opt,opt,...*`)`
: Sets privacy options. Privacy is really a misnomer; many of these options are just a way of insisting on stricter adherence to the SMTP protocol.

  The `goaway` pseudo-flag sets all flags except `noreceipts`, `restrictmailq`, `restrictqrun`, `restrictexpand`, `noetrn`, and `nobodyreturn`. If `mailq` is restricted, only people in the same group as the queue directory can print the queue. If queue runs are restricted, only root and the owner of the queue directory can run the queue. The `restrict-expand` pseudo-flag instructs sendmail to drop privileges when the -bv option is given by users who are neither root nor the `TrustedUser` so users cannot read private aliases, forwards, or `:include:` files. It adds the `NonRootSafeAddr` to the "DontBlame-Sendmail" option to prevent misleading unsafe address warnings. It also overrides the `-v` (verbose) command line option to prevent information leakage. Authentication Warnings add warnings about various conditions that may indicate attempts to fool the mail system, such as using an non-standard queue directory.

  The options can be selected from:

  `authwarnings`
  : Puts `X-Authentication-Warning:` headers in messages.

  `goaway`
  : Disallows essentially all SMTP status queries.

  `needexpnhelo`
  : Insists on `HELO` or `EHLO` command before `EXPN`.

  `needmailhelo`
  : Insists on `HELO` or `EHLO` command before `MAIL`.

  `needvrfyhelo`
  : Insists on `HELO` or `EHLO` command before `VRFY`.

  `noactualrecipient`
  : Do not put an X-Actual-Recipient line in a DNS that reveals the actual account to which an address is mapped.

  `noetrn`
  : Disallows `ETRN` entirely.

  `noexpn`
  : Disallows `EXPN` entirely.

`noreceipts`
Prevents return receipts.

`nobodyreturn`
Does not return the body of a message with DSNs.

`novrfy`
Disallows `VRFY` entirely.

`public`
Allows open access.

`restrictexpand`
Restricts `-bv` and `-v` command line flags.

`restrictmailq`
Restricts `mailq` command.

`restrictqrun`
Restricts `-q` command line flag.

`ProcessTitlePrefix` *string*
Prefixes the process title shown on "`/usr/ucb/ps auxww`" listings with *string*. The string is macro processed.

`QueueDirectory (Q`*dir*`)`
Uses the named *dir* as the queue directory.

`QueueFactor (q`*factor*`)`
Uses *factor* as the multiplier in the map function to decide when to just queue up jobs rather than run them. This value is divided by the difference between the current load average and the load average limit (`x` flag) to determine the maximum message priority to be sent. Defaults to `600000`.

`QueueFileMode=`*mode*
Defaults permissions for queue files (octal). If not set, `sendmail` uses `0600` unless its real and effective `uid` are different in which case it uses `0644`.

`QueueLA (x`*LA*`)`
When the system load average exceeds *LA*, just queues messages (that is, does not try to send them). Defaults to eight times the number of processors online when `sendmail` starts.

`QueueSortOrder=`*algorithm*
Sets the algorithm used for sorting the queue. Only the first character of the value is used. Legal values are `host` (to order by the name of the first host name of the first recipient), `filename` (to order by the name of the queue file name), `time` (to order by the submission/creation time), `random` (to order randomly), `modification` (to order by the modification time of the `qf` file (older entries first)), `none` (to not order), and `priority` (to order by message priority). Host ordering makes better use of the connection cache, but may tend to process low priority messages that go to a single host over high priority

messages that go to several hosts; it probably shouldn't be used on slow network links. Filename and modification time ordering saves the overhead of reading all of the queued items before starting the queue run. Creation (submission) time ordering is almost always a bad idea, since it allows large, bulk mail to go out before smaller, personal mail, but may have applicability on some hosts with very fast connections. Random is useful if several queue runners are started by hand which try to drain the same queue since odds are they are working on different parts of the queue at the same time. Priority ordering is the default.

`QueueTimeout (T`*rtime*`/`*wtime*`)`
Sets the queue timeout to *rtime*. After this interval, messages that have not been successfully sent are returned to the sender. Defaults to five days (`5d`). The optional *wtime* is the time after which a warning message is sent. If it is missing or `0`, then no warning messages are sent.

`RandFile`
File containing random data (use prefix `file:`) or the name of the UNIX socket if EGD is used (use prefix `egd:`). Note that Solaris supports `random(7D)`, so this does not need to be specified.

`RecipientFactor (y`*fact*`)`
The indicated factor *fact* is added to the priority (thus *lowering* the priority of the job) for each recipient, that is, this value penalizes jobs with large numbers of recipients. Defaults to `30000`.

`RefuseLA (X`*LA*`)`
When the system load average exceeds *LA*, refuses incoming SMTP connections. Defaults to 12 times the number of processors online when `sendmail` starts.

`RejectLogInterval`
Log interval when refusing connections for this long (default: 3h).

`ResolverOptions (I)`
Tunes DNS lookups.

`RetryFactor (Z`*fact*`)`
The indicated factor *fact* is added to the priority every time a job is processed. Thus, each time a job is processed, its priority is decreased by the indicated value. In most environments this should be positive, since hosts that are down are all too often down for a long time. Defaults to `90000`.

`RrtImpliesDsn`
If this option is set, a `Return-Receipt-To`: header causes the request of a DSN, which is sent to the envelope sender as required by RFC 1891, not to the address given in the header.

`RunAsUser`
If set, becomes this user when reading and delivering mail. Intended for use of firewalls where users do not have accounts.

`SafeFileEnvironment`
: If set, `sendmail` does a `chroot` into this directory before writing files.

`SaveFromLine (f)`
: Saves Unix-style `From` lines at the front of headers. Normally they are assumed redundant and discarded.

`SendMimeErrors (j)`
: If set, sends error messages in MIME format (see RFC 2045 and RFC 1344 for details). If disabled, `sendmail` does not return the DSN keyword in response to an `EHLO` and does not do Delivery Status Notification processing as described in RFC 1891.

`ServerCertFile`
: File containing the cert of the server, that is, this cert is used when `sendmail` acts as server.

`ServerKeyFile`
: File containing the private key belonging to the server cert.

`ServiceSwitchFile`
: Defines the path to the service-switch file. Since the service-switch file is defined in the Solaris operating environment this option is ignored.

`SevenBitInput (7)`
: Strips input to seven bits for compatibility with old systems. This should not be necessary.

`SharedMemoryKey`
: Specifies key to use for shared memory segment. If not set (or `0`), shared memory is not be used. If this option is set, `sendmail` can share some data between different instances. For example, the number of entries in a queue directory or the available space in a file system. This allows for more efficient program execution, since only one process needs to update the data instead of each individual process gathering the data each time it is required.

`SharedMemoryKeyFile=`*file*
: If `SharedMemoryKeyFile` is set to `-1`, the automatically selected shared memory key will be stored in the specified file.

`SingleLineFromHeader`
: If set, `From:` lines that have embedded newlines are unwrapped onto one line.

`SingleThreadDelivery`
: If this option and the `HostStatusDirectory` option are both set, uses single thread deliveries to other hosts.

`SmtpGreetingMessage or $e`
: Specifies the initial SMTP greeting message.

`SoftBounce`
: If set, issue temporary errors (`4xy`) instead of permanent errors (`5xy`). This can be useful during testing of a new configuration to avoid erroneous bouncing of mail.

`StatusFile (S`*file*`)`
Logs statistics in the named *file*. By default, this is `/etc/mail/sendmail.st`. As root, you must `touch(1)` this file to enable `mailstats(1)`.

`SuperSafe (s)`
This option can be set to `True`, `False`, `Interactive`, or `PostMilter`. If set to `True`, `sendmail` is set to super-safe when running things, that is, always instantiate the queue file, even if you are going to attempt immediate delivery. `sendmail` always instantiates the queue file before returning control to the client under any circumstances. This should really always be set to `True`. The `Interactive` value has been introduced in `8.12` and can be used together with `DeliveryMode=i`. It skips some synchronization calls which are effectively doubled in the code execution path for this mode. If set to `PostMilter`, `sendmail` defers synchronizing the queue file until any milters have signaled acceptance of the message. `PostMilter` is useful only when `sendmail` is running as an SMTP server; in all other situations it acts the same as True.

`TempFileMode (F`*mode*`)`
Specifies the file mode for queue files.

`Timeout (r`*timeouts*`)`
Timeout reads after time interval. The *timeouts* argument is a list of *keyword*=*value* pairs. All but *command* apply to client SMTP. For backward compatibility, a timeout with no *keyword*= part is set all of the longer values. The recognized timeouts and their default values, and their minimum values specified in RFC 1123 section 5.3.2 are:

`aconnect`
all connections for a single delivery attempt [0, unspecified]

`command`
command read [1h, 5m]

`connect`
initial connect [0, unspecified]

`control`
complete control socket transaction [2m, none]

`datablock`
data block read [1h, 3m]

`datafinal`
reply to final `.` in data [1h, 10m]

`datainit`
reply to `DATA` command [5m, 2m]

`fileopen`
file open [60sec, none]

`helo`
reply to `HELO` or `EHLO` command [5m, none]

`hoststatus`
: host retry [30m, unspecified]

`iconnect`
: first attempt to connect to a host [0, unspecified]

`ident`
: `IDENT` protocol timeout [5s, none]

`initial`
: wait for initial greeting message [5m, 5m]

`lhlo`
: wait for reply to an LMTP LHLO command [2m, unspecified]

`mail`
: reply to `MAIL` command [10m, 5m]

`misc`
: reply to `NOOP` and `VERB` commands [2m, none]

`queuereturn`
: undeliverable message returned [5d]

`queuewarn`
: deferred warning [4h]

`quit`
: reply to `QUIT` command [2m, none]

`rcpt`
: reply to `RCPT` command [1h, 5m]

`resolver.retrans`
: Resolver's retransmission time interval (in seconds) [varies]. Sets both `Timeout.resolver.retrans.first` and `Timeout.resolver.retrans.normal`.

`resolver.retrans.first`
: Resolver's retransmission time interval (in seconds) for the first attempt to deliver a message [varies].

`resolver.retrans.normal`
: Resolver's retransmission time interval (in seconds) for all look-ups except the first delivery attempt [varies].

`resolver.retry`
: Number of times to retransmit a resolver query [varies]. Sets both `Timeout.resolver.retry.first` and `Timeout.resolver.retry.normal`.

`resolver.retry.first`
: Number of times to retransmit a resolver query for the first attempt to deliver a message [varies].

`resolver.retry.normal`
: Number of times to retransmit a resolver query for all look-ups except the first delivery attempt [varies].

`rset`
: reply to `RSET` command [5m, none]

`starttls`
: response to an `SMTP` `STARTTLS` command [1h]

`TimeZoneSpec (t`*tzinfo*`)`
: Sets the local time zone info to *tzinfo*, for example, "PST8PDT". Actually, if this is not set, the `TZ` environment variable is cleared (so the system default is used); if set but null, the user's `TZ` variable is used, and if set and non-null, the `TZ` variable is set to this value.

`TLSSrvOptions`
: If this option is '`V`', then no client verification is performed,that is, the server does not ask for a certificate.

`TrustedUser`
: The user parameter can be a user name (looked up in the passwd map) or a numeric user id. Trusted user for file ownership and starting the daemon. If set, generated alias databases and the control socket (if configured) are automatically owned by this user.

`TryNullMXList (w)`
: If you are the "best" (that is, lowest preference) MX for a given host, you should normally detect this situation and treat that condition specially, by forwarding the mail to a UUCP feed, treating it as local, or whatever. However, in some cases (such as Internet firewalls) you may want to try to connect directly to that host as though it had no MX records at all. Setting this option causes `sendmail` to try this. The downside is that errors in your configuration are likely to be diagnosed as "host unknown" or "message timed out" instead of something more meaningful. This option is deprecated.

`UnixFromLine or $l`
: The "From " line used when sending to files or programs.

`UnsafeGroupWrites`
: If set, group-writable :include: and .forward files are considered "unsafe", that is, programs and files cannot be directly referenced from such files.

`UseErrorsTo (l)`
: If there is an `Errors-To`: header, sends error messages to the addresses listed there. They normally go to the envelope sender. Use of this option causes `sendmail` to violate RFC 1123. This option is not recommended and deprecated.

`UseMSP`
: Uses as mail submission program, that is, allows group writable queue files if the group is the same as that of a set-group-id `sendmail` binary.

`UserDatabaseSpec (U)`
: Defines the name and location of the file containing User Database information.

`Verbose (v)`
: Runs in verbose mode. If this is set, `sendmail` adjusts the `HoldExpensive` and `DeliveryMode` options so that all mail is delivered completely in a single job so that you can see the entire delivery process. The `Verbose` option should *never* be set in the configuration file; it is intended for command line use only.

`XscriptFileBufferSize`
: Sets the threshold, in bytes, before a memory-bases queue transcript file becomes disk-based. The default is 4096 bytes.

If the first character of the user name is a vertical bar, the rest of the user name is used as the name of a program to pipe the mail to. It may be necessary to quote the name of the user to keep `sendmail` from suppressing the blanks from between arguments.

If invoked as `newaliases`, `sendmail` rebuilds the alias database, so long as the `/etc/mail/aliases*` files are owned by root *and* root has exclusive write permission. If invoked as `mailq`, `sendmail` prints the contents of the mail queue.

オペランド

address
: address of an intended recipient of the message being sent.

使用法

See `largefile(5)` for the description of the behavior of `sendmail` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

終了ステータス

`sendmail` returns an exit status describing what it did. The codes are defined in `/usr/include/sysexits.h`.

`EX_OK`
: Successful completion on all addresses.

`EX_NOUSER`
: User name not recognized.

`EX_UNAVAILABLE`
: Catchall. Necessary resources were not available.

`EX_SYNTAX`
: Syntax error in address.

`EX_SOFTWARE`
: Internal software error, including bad arguments.

`EX_OSERR`
: Temporary operating system error, such as “cannot fork”.

`EX_NOHOST`
: Host name not recognized.

`EX_TEMPFAIL`
Message could not be sent immediately, but was queued.

環境変数

No environment variables are used. However, sendmail's start-up script, invoked by `svcadm(1M)`, reads `/etc/default/sendmail`. In this file, if the variable `ETRN_HOSTS` is set, the start-up script parses this variable and invokes `etrn(1M)` appropriately. `ETRN_HOSTS` should be of the form:

```
"s1:c1.1,c1.2        s2:c2.1 s3:c3.1,c3.2,c3.3"
```

That is, white-space separated groups of *server:client* where *client* can be one or more comma-separated names. The *:client* part is optional. *server* is the name of the server to prod; a mail queue run is requested for each *client* name. This is comparable to running:

```
/usr/lib/sendmail -qR client
```

on the host *server*.

ファイル

`dead.letter`
Unmailable text

`/etc/default/sendmail`
Contains default settings. You can override some of the settings by command line options.

`/etc/mail/aliases`
Mail aliases file (ASCII)

`/etc/mail/aliases.db`
Database of mail aliases (binary)

`/etc/mail/aliases.dir`
Database of mail aliases (binary)

`/etc/mail/aliases.pag`
Database of mail aliases (binary)

`/etc/mail/sendmail.cf`
Defines environment for `sendmail`

`/etc/mail/submit.cf`
Defines environment for MSP

`/etc/mail/trusted-users`
Lists users that are “trusted”, that is, able to set their envelope from address using `-f` without generating a warning message. Note that this file is consulted by the default `sendmail.cf`, but not by the default `submit.cf`, in which the line referring to `/etc/mail/trusted-users` is commented out. See `sendmail(4)` for instructions on making changes to `submit.cf` and `sendmail.cf`.

`/var/spool/clientmqueue/*`
Temporary files and queued mail

`/var/spool/mqueue/*`
: Temporary files and queued mail

`~/.forward`
: List of recipients for forwarding messages

`/usr/include/libmilter/README`
: Describes the steps needed to compile and run a filter

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

関連項目

`svcs(1)`, `biff(1B)`, `mail(1)`, `mailq(1)`, `mailx(1)`, `nice(1)`, `check-hostname(1M)`, `check-permissions(1M)`, `etrn(1M)`, `newaliases(1M)`, `svcadm(1M)`, `svccfg(1M)`, `fork(2)`, `getpwnam(3C)getusershell(3C)resolver(3RESOLV)aliases(4)`, `hosts(4)`, `sendmail(4)`, `shells(4)`, `attributes(5)`, `largefile(5)`, `smf(5)`, `random(7D)`

`tcpd`(1M), `hosts_access`(4) in the `security/tcp-wrapper` package.

RFC 2821 Simple Mail Transfer Protocol, John Klensin, April 2001.

RFC 2822 Internet Message Format, Pete Resnick, April 2001.

sendmail, Third Edition, Bryan Costales with Eric Allman, O'Reilly & Associates, Inc., 2003.

http://www.sendmail.org

http://www.milter.org

注意事項

The `sendmail` program requires a fully qualified host name when starting. A script has been included to help verify if the host name is defined properly (see `check-hostname(1M)`).

The permissions and the ownership of several directories have been changed in order to increase security. In particular, access to `/etc/mail` and `/var/spool/mqueue` has been restricted.

Security restrictions have been placed users using `.forward` files to pipe mail to a program or redirect mail to a file. The default shell (as listed in `/etc/passwd`) of these users must be listed in `/etc/shells`. This restriction does not affect mail that is being redirected to another alias.

Additional restrictions have been put in place on `.forward` and `:include:` files. These files and the directory structure that they are placed in cannot be group- or world-writable. See `check-permissions(1M)`.

If you have interfaces that map to domains that have MX records that point to non-local destinations, you might need to enable the `DontProbeInterfaces` option to enable delivery to those destinations. In its default startup behavior, `sendmail` probes each interface and adds an

interface's IP addresses, as well as any domains that those addresses map to, to its list of domains that are considered local. For domains thus added, being on the list of local domains is equivalent to having a 0-preference MX record, with `localhost` as the MX value. If this is not the result you want, enable `DontProbeInterfaces`.

名前

sf880drd – Sun Fire 880 Dynamic Reconfiguration daemon

形式

`sf880drd`

機能説明

The Sun Fire 880 Dynamic Reconfiguration daemon, `sf880drd`, is part of the PCI and system bus hotplug framework. `sf880drd` starts at boot time. It has no configuration options and does not report any system status.

`sf880drd` implements the Sun Fire `880` console-less system administration (per-slot pushbuttons and LED status indicators). It also manages various aspects of CPU/memory hotplug.

ファイル

`/usr/platform/SUNW,Sun-Fire-880/lib/sf880drd`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/kernel/dynamic-reconfiguration/sun-fire-880

関連項目

`svcs(1)`, `cfgadm(1M)`, `cfgadm_pci(1M)`, `cfgadm_sbd(1M)`, `svcadm(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `sf880drd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/platform/sun4u/sf880drd
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

sftp-server – SFTP server subsystem

形式

`/usr/lib/ssh/sftp-server [-f log_facility] [-l log_level]`

機能説明

`sftp-server` implements the server side of the SSH File Transfer Protocol as defined in the IETF `draft-ietf-secsh-filexfer`.

`sftp-server` is a subsystem for `sshd(1M)` and must not be run directly. Command-line flags to `sftp-server` should be specified in the Subsystem declaration. See `sshd_config(4)` for more information.

To enable the `sftp-server` subsystem for `sshd` add the following to `/etc/ssh/sshd_config`:

```
Subsystem   sftp     /usr/lib/ssh/sftp-server
```

To run `sftp-server` in a `chroot` configuration, use `internal-sftp` instead of `/usr/lib/ssh/sftp-server`. Otherwise, the `chroot` directory must contain the necessary files and directories to support the user's session. See the `ChrootDirectory` and `Subsystem` options in `sshd_config(4)`) for more information on how `sshd` and `sftp-server` work with `chroot(2)`.

See `sshd_config(4)` for a description of the format and contents of that file.

There is no relationship between the protocol used by `sftp-server` and the FTP protocol (RFC 959) provided by `in.ftpd`.

For logging to work, `sftp-server` must be able to access `/dev/log`. Use of `sftp-server` in a `chroot` configuration therefore requires that `syslogd(1M)` establish a logging socket inside the `chroot` directory.

オプション

Valid options are listed below. As stated above, these options, if used, are specified in the Subsystem declaration of `sshd_config`.

`-f` *log_facility*
: Specifies the facility code that is used when logging messages from `sftp-server`. The possible values are: `DAEMON`, `USER`, `AUTH`, `LOCAL0`, `LOCAL1`, `LOCAL2`, `LOCAL3`, `LOCAL4`, `LOCAL5`, `LOCAL6`, `LOCAL7`. The default is `AUTH`.

`-l` *log_level*
: Specifies which messages will be logged by `sftp-server`. The possible values are: `QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, `DEBUG`, `DEBUG1`, `DEBUG2`, and `DEBUG3`. `INFO` and `VERBOSE` log transactions that `sftp-server` performs on behalf of the client. `DEBUG` and `DEBUG1` are equivalent. `DEBUG2` and `DEBUG3` each specify higher levels of debugging output. The default is `ERROR`.

`-u` *umask*
: Sets an explicit `umask(2)` to be applied to newly-created files and directories, instead of the user's default mask.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/usr/lib/ssh/sftp-server` Server-side binary.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/ssh
Interface Stability	Committed

関連項目

`sftp(1)`, `ssh(1)`, `ssh-add(1)`, `ssh-keygen(1)`, `sshd(1M)`, `syslogd(1M)`, `chroot(2)`, `umask(2)`, `sshd_config(4)`, `attributes(5)`

名前

shadowd – shadow migration daemon

形式

```
/usr/lib/fs/shadowd
```

```
svc:/system/filesystem/shadowd:default
```

機能説明

shadowd is a daemon that provides background worker threads to migrate data for a shadow migration. A shadow migration gradually moves data from a source file system into a new "shadow" file system. Users can access and change their data within the shadow file system while migration is occurring.

The shadowd service is managed by the service management facility, smf(5). Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

The svccfg(1M) command can be used to manage the following parameter related to shadowd:

```
config_params/shadow_threads
```

Number of threads to devote to background migration of data. These threads are global to the entire system, and increasing the number can increase concurrency and the overall speed of migration at the expense of increased resource consumption (network, I/O, and CPU).

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/zfs
Interface Stability	Unstable

関連項目

shadowstat(1M), zfs(1M), attributes(5)

名前

shadowstat – report shadow migration statistics

形式

`/usr/sbin/shadowstat [count]`

機能説明

The `shadowstat` utility reports statistics on in-progress shadow migrations. A line of output is presented for each migrating file system; the output looks similar to the following:

```
                        EST
                BYTES   BYTES           ELAPSED
DATASET         XFRD    LEFT    ERRORS  TIME
pool/newfs      2.10M   28.8G   -       00:02:45
```

Each line identifies bytes transferred thus far, a rough estimate of bytes left to transfer, number of migration errors, and the time elapsed since the migration was started.

If *count* is unspecified, updated values are presented every few seconds until the program is terminated. If *count* is specified, updates only occur *count* times, after which the program exits.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/zfs
Interface Stability	Unstable

関連項目

`shadowd(1M)`, `zfs(1M)`, `attributes(5)`

名前

share – ファイルシステム共有を表示する、またはローカルファイルシステムをリモートシステムからマウント可能にする

形式

share [-F *protocol*] -a

share [-F *protocol*] [-o *options*] [-d *description*] *pathname* [*sharename*]

share [-F *protocol*] [-A]

機能説明

share コマンドは、ファイルシステムの共有を定義および公開します。これは、共有プロトコルを使用してファイルシステムをマウントできることを意味します。

-F *protocol* オプションを指定しない場合、/etc/dfs/fstypes 内の最初のファイル共有プロトコルがデフォルトとして使用されます。

NFS に固有の共有オプションについては、share_nfs(1M) を参照してください。SMB に固有の共有オプションについては、share_smb(1M) を参照してください。

share コマンドを使用した、ZFS ファイルシステムの NFS 共有または SMB 共有の定義および公開は、古い運用方法であるとみなされています。zfs set share コマンドを使用して NFS または SMB の共有プロパティーを定義し、ZFS の sharenfs または sharesmb プロパティーを設定して共有を公開することを検討してください。詳細は、zfs(1M) および share_smb(1M) を参照してください。

上記の「形式」に示された share コマンドの 3 つ目の形式では、share は公開された共有を表示するか、-A オプションを付けてすべての構成済み (定義済み) の共有を表示します。

オプション

-F *protocol*
: ファイル共有プロトコルを指定します。

-o *specific_options*

rw
: 共有 *pathname* は、すべてのクライアントに読み取りおよび書き込み権を付与して公開されます。これはデフォルトの動作です。

rw=*client*[:*client*]...
: 共有 *pathname* は、リストされたクライアントにのみ読み取りおよび書き込み権を付与して公開されます。ほかのシステムは共有パス名にアクセスできません。

ro
: 共有 pathname は、すべてのクライアントに読み取り専用アクセス権を付与して公開されます。

ro=*client*[:*client*]...
: 共有 pathname は、リストされたクライアントにのみ読み取り専用アクセス権を付与して公開されます。ほかのシステムは共有パス名にアクセスできません。

複数のオプションはコンマで区切ります。オプションごとの複数のオペランドはコロンで区切ります。「使用例」を参照してください。

`-d`
公開されるファイルシステムの共有について説明したコメントを指定します。

`-a`
定義されているすべての共有を公開します。

`-A`
定義されているすべての共有を表示します。

使用例

例 1　読み取り専用アクセス権を付与した NFS 共有の公開

次のコマンドは、読み取り専用アクセス権を付与して `/ufsfs` の NFS 共有を定義および公開します。

```
# share -F nfs -o ro /ufsfs
```

例 2　複数の共有オプションを指定した NFS 共有の公開

次のコマンドは、読み取り専用アクセス権を持つ `users_nfs` という `netgroup`、および読み取りおよび書き込み権を持つ指定されたホストのユーザーを指定して、`/export/manuals` ファイルシステムの NFS 共有を定義および公開します。

```
# share -F nfs -o ro=users_nfs,rw=host1:host2:host3 /export/manuals
```

ファイル

`/etc/dfs/dfstab`
このファイルは廃止されています。SMF サービスは、NFS 共有または SMB 共有をブート時に公開します。

`/etc/dfs/fstypes`
ファイル共有プロトコルのリスト。デフォルトのファイル共有プロトコルは NFS です。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	SUNWcs

関連項目

`mountd(1M)`, `nfsd(1M)`, `share_nfs(1M)`, `share_smb(1M)`, `shareall(1M)`, `unshare(1M)`, `zfs(1M)`, `attributes(5)`

注意事項

`share` コマンドが同じファイルシステムに対して複数回呼び出された場合、最後の `share` の呼び出しが前の呼び出しより優先されます。最後の `share` コマンドによって設定されたオプションによって、古いオプションが置き換えられます。たとえば、`/somefs` の `usera` に読み取りおよび書き込み権が付与されていて、`/somefs` の `userb` にも読み取りおよび書き込み権を付与する場合は、次の構文を使用します。

```
example% share -F nfs -o rw=usera:userb /somefs
```

名前

shareall, unshareall – 複数の共有を公開または非公開にする

形式

`shareall [-F FSType [,FSType]...] [-| file]`

`unshareall [-F FSType [,FSType]...]`

機能説明

引数を指定しないで使用すると、shareall は、共有コマンド行のリストを含む指定されたファイルから共有を公開します。オペランドにハイフン (-) を指定すると、share コマンド行は標準入力から読み取られます。あるいは、ファイルもハイフンも指定されていない場合、共有は SMF 共有リポジトリから公開されます。

共有は、ファイルシステムタイプをコンマで区切られたリストに記述して、-F の引数にすることによって指定できます。

unshareall は、公開されているファイルシステム共有をすべて非公開にします。-F フラグがない場合は、すべての分散ファイルシステムの共有を非公開にします。

オプション

-F *FSType*　ファイルシステムを指定します。デフォルトは、/etc/dfs/fstypes 内の最初のエントリです。

ファイル

/etc/dfs/dfstab

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	SUNWcs

関連項目

share(1M), unshare(1M), attributes(5)

名前

sharectl – ファイル共有サービスの構成および管理

形式

```
sharectl [-h]

sharectl status [-h] [protocol]

sharectl get [-h] [-p property]... protocol

sharectl set [-h] [-p property=value]... protocol
```

機能説明

sharectl コマンドは、NFS や SMB などのファイル共有プロトコルに対して処理を実行します。このコマンドは、クライアントおよびサーバー操作プロパティーを設定し、構成スナップショットを取得および復元し、プロトコルサービスのステータスを取得します。

get および set サブコマンド (下記参照) では、root 特権を持っているか、または ユーザーが Primary Administrator 役割を引き受けていることが必要です。承認 ユーザーは、sharectl を使用すると、Solaris サーバー管理機能で NFS および SMB プロパティーのグローバル値を設定できます。nfs(4) および smb(4) を参照してください。

オプション

次のオプションがサポートされています。

-h
: 使用方法に関するメッセージを表示します。

-p *property*[= *value*]
: プロパティーを指定します。後述の「サブコマンド」を参照してください。

サブコマンド

sharectl は、次のサブコマンドをサポートします。sharectl コマンドの形式は次のとおりです。

```
# sharectl subcommand [option]
```

get [-p *property*] *protocol*	指定されたプロトコルのプロパティー値を取得します。-p オプションが指定されていない場合、指定されたプロトコルのすべてのプロパティーを取得します。
set [-p *property=value*]... *protocol*	指定されたファイル共有プロトコルのプロパティーを設定します。
status [*protocol*]	指定されたプロトコルのステータスを表示します。プロトコルが指定されていない場合は、すべてのファイル共有プロトコルのステータスを表示します。

使用例

例 1　プロパティーを取得する

次のコマンドは、NFS プロトコルのプロパティーを取得します。

例 1　プロパティーを取得する　　*(続き)*

```
% sharectl get nfs
lockd_listen_backlog=32
lockd_servers=20
lockd_retransmit_timeout=5
grace_period=90
nfsmapid_domain=sun.com
servers=16
server_versmin=2
server_versmax=4
max_connections=-1
```

次のコマンドは、NFS プロトコルの grace_period プロパティーの値を取得します。

```
% sharectl get -p grace_period nfs
grace_period=90
```

例 2　プロパティーを設定する

前述の例では、サーバー NFS プロトコル (`server_versmin`) の最小バージョンが、2 に設定されています。次のコマンドは、最低バージョン番号をバージョン 3 に設定します。

```
% sharectl set -p server_versmin=3 nfs
```

例 3　ステータスを取得する

次のコマンドは、システムでのすべてのファイル共有プロトコルのステータスを取得します。

```
% sharectl status
nfs      enabled
```

例 4　SMB サーバーのプロパティーを設定する

次のコマンドは、SMB プロトコルの `server_signing_required` プロパティー値を設定します。

```
% sharectl set -p server_signing_required=true smb
```

例 5　SMB クライアントのプロパティーを設定する

次のコマンドは、SMB プロトコルの `client_signing_required` プロパティー値を設定します。

```
% sharectl set -p client_signing_required=true smb
```

例 6 autofs の RPC 呼び出しのトレースを設定する

次のコマンドは、各 RPC 呼び出しを autofs に展開し、automountd(1M) でそのサービスに対して指定された場所に記録します。

```
# sharectl set trace=1 autofs
```

終了ステータス

0 正常終了。

0 以外 コマンドが失敗しました。

ファイル

/usr/include/libshare.h 終了ステータスに使用されるエラーコード。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

automount(1M), automountd(1M), lockd(1M), mountd(1M), nfsd(1M), nfsmapid(1M), nfs(4), smb(4), user_attr(4), attributes(5), rbac(5), smf(5), standards(5)

名前

share_nfs – make NFS shares available for mounting by remote systems

形式

```
share -F nfs [-a [-o specific_options] [-d description]
     pathname [sharename] | [-A]]
zfs set share=name=share-name,path=pathname,[desc="description"],
     prot=nfs,[[property=value[,...]] dataset
```

機能説明

The `share` utility defines and publishes a NFS share, which makes a local file system available for mounting by remote systems. It starts the `nfsd(1M)` and `mountd(1M)` daemons if they are not already running.

You can use the `share` command to create and publish a ZFS file system share, but this is considered a legacy operation. See `zfs(1M)` for information about using the `zfs set share` command and the `sharenfs` property to create and publish NFS shares.

オプション

The following options are supported:

`-F nfs`
: Specify the NFS file sharing protocol.

`-a`
: Publish all defined shares.

`-o` *specific_options*
: Specify *specific_options* in a comma-separated list of keywords and attribute-value-assertions for interpretation by the NFS protocol. By default, a share is published with read-write access to all clients, unless a specific option overrides the default access. *specific_options* can be any combination of the following:

  `aclok`
  : Allows the NFS server to do access control for NFS Version 2 clients. When `aclok` is set on the server, maximal access is given to all clients. For example, with `aclok` set, if anyone has read permissions, then everyone does. If `aclok` is not set, minimal access is given to all clients.

  `anon=`*uid*
  : Set *uid* to be the effective user ID of unknown users. By default, unknown users are given the effective user ID `UID_NOBODY`. If *uid* is set to `–1`, access is denied.

  charset
  : All clients will be assumed to be using the specified character set (see list in following description) and file and path names will be converted to UTF-8 for the server.

  charset=access_list
  : Where *charset* is one of: `euc-cn`, `euc-jp`, `euc-jpms`, `euc-kr`, `euc-tw`, `iso8859-1`, `iso8859-2`, `iso8859-5`, `iso8859-6`, `iso8859-7`, `iso8859-8`, `iso8859-9`, `iso8859-13`, `iso8859-15`, `koi8-r`.

    Clients that match the *access_list* for one of these properties will be assumed to be using that character set and file and path names will be converted to UTF-8 for the server.

`index=file`
: Load `file` rather than a listing of the directory containing this file when the directory is referenced by an NFS URL.

`log=`*tag*
: Enables NFS server logging for the specified file system. The optional tag determines the location of the related log files. The *tag* is defined in `etc/nfs/nfslog.conf`. If no *tag* is specified, the default values associated with the *global tag* in `etc/nfs/nfslog.conf` is used. Support of NFS server logging is only available for NFS Version 2 and Version 3 requests.

`noaclfab`
: Allows NFS servers to not return fabricated ACLs to NFS clients. The default behavior for NFS servers is to fabricate ACLs. If `noaclfab` is set, then the NFS server does not fabricate ACLs, which is the appropriate choice if the underlying filesystem does not support the POSIX Draft ACL.

`none`
: Access is disallowed to all clients. The `ro` or `rw` options can override `none`.

`none=`*access_list*
: Access is not allowed to any client that matches the access list. The exception is when the access list is an asterisk (`*`), in which case `ro` or `rw` can override `none`.

`nosub`
: Prevents clients from mounting subdirectories of shared directories. For example, if `/export` is shared with the `nosub` option on server *fooey* then a NFS client cannot do:

```
mount -F nfs fooey:/export/home/mnt
```

NFS Version 4 does not use the `MOUNT` protocol. The `nosub` option only applies to NFS Version 2 and Version 3 requests.

`nosuid`
: By default, clients are allowed to create files on the shared file system with the setuid or setgid mode enabled. Specifying `nosuid` causes the server file system to silently ignore any attempt to enable the setuid or setgid mode bits.

`public`
: Moves the location of the public file handle from `root` (`/`) to the exported directory for WebNFS-enabled browsers and clients. This option does not enable WebNFS service. WebNFS is always on. Only one file system per server may use this option. Any other option, including the `-ro=list` and `-rw=list` options can be included with the `public` option.

`ro`
: Share is published with read-only access to all clients.

`ro=`*access_list*
: Share is published with read-only access to the clients listed in *access_list*; overrides the `rw` suboption for the clients specified. See *access_list* below.

`root`
: Root users from all hosts have root access.

`root=`*access_list*
: Only root users from the hosts specified in *access_list* have root access. See *access_list* below. By default, no host has root access, so root users are mapped to an anonymous user ID (see the `anon=`*uid* option described above). Netgroups can be used if the file system shared is using UNIX authentication (`AUTH_SYS`).

`root_mapping=`*uid*
: For a client that is allowed root access, map the root UID to the specified user id.

`rw`
: Share is published with read and write access to all clients.

`rw=`*access_list*
: Share is published with read and write access to the clients listed in *access_list*; overrides the `ro` suboption for the clients specified. See *access_list* below.

`sec=`*mode*[`:`*mode*]. . .
: Publishes a share by using one or more of the specified security modes. The *mode* in the `sec=`*mode* option must be a node name supported on the client. If the `sec=` option is not specified, the default security mode used is `AUTH_SYS`. Multiple `sec=` options can be specified on the command line, although each mode can appear only once. The security modes are defined in `nfssec(5)`.

  Each `sec=` option specifies modes that apply to any subsequent `window=, rw, ro, rw=, ro=` and `root=` options that are provided before another `sec=`option. Each additional `sec=` resets the security mode context, so that more `window=, rw, ro, rw=, ro=` and `root=` options can be supplied for additional modes.

`sec=`*none*
: If the option `sec=`*none* is specified when the client uses `AUTH_NONE`, or if the client uses a security mode that is not one that the file system is shared with, then the credential of each NFS request is treated as unauthenticated. See the `anon=`*uid* option for a description of how unauthenticated requests are handled.

`secure`
: This option has been deprecated in favor of the `sec=`*dh* option.

`window=`*value*
: When a share is published with `sec=`*dh*, set the maximum life time (in seconds) of the RPC request's credential (in the authentication header) that the NFS server allows. If a credential arrives with a life time larger than what is allowed, the NFS server rejects the request. The default value is 30000 seconds (8.3 hours).

-d *description*
: Provide a comment that describes the file system to be shared.

-A
: Display all defined shares.

access_list

The *access_list* argument is either the string "*" to represent all hosts or a colon-separated list whose components may be any number of the following:

hostname
: The name of a host. With a server configured for DNS or LDAP naming in the `nsswitch` "hosts" entry, any hostname must be represented as a fully qualified DNS or LDAP name. The hostname specified must be the canonical name for this host and must match the hostname returned on the reverse lookup of the incoming IP address of the NFS client.

netgroup
: A netgroup contains a number of hostnames. With a server configured for DNS or LDAP naming in the `nsswitch` "hosts" entry, any hostname in a netgroup must be represented as a fully qualified DNS or LDAP name.

domain name suffix
: To use domain membership the server must use DNS or LDAP to resolve hostnames to IP addresses; that is, the "hosts" entry in the `/etc/nsswitch.conf` must specify "dns" or "ldap" ahead of "nis", since only DNS and LDAP return the full domain name of the host. Other name services like NIS cannot be used to resolve hostnames on the server because when mapping an IP address to a hostname they do not return domain information. For example,

```
NIS   172.16.45.9 --> "myhost"
```

and

```
DNS or LDAP   172.16.45.9 -->
     "myhost.mydomain.mycompany.com"
```

The domain name suffix is distinguished from hostnames and netgroups by a prefixed dot. For example,

```
rw=.mydomain.mycompany.com
```

A single dot can be used to match a hostname with no suffix. For example,

```
rw=.
```

matches "mydomain" but not "mydomain.mycompany.com". This feature can be used to match hosts resolved through NIS rather than DNS and LDAP.

network
: The network or subnet component is preceded by an at-sign (@). It can be either a name or a dotted address. If a name, it is converted to a dotted address by `getnetbyname`(3SOCKET). For example,

```
=@mynet
```

would be equivalent to:

`=@172.16` or `=@172.16.0.0`

The network prefix assumes an octet-aligned netmask determined from the zeroth octet in the low-order part of the address up to and including the high-order octet, if you want to specify a single IP address (see below). In the case where network prefixes are not byte-aligned, the syntax allows a mask length to be specified explicitly following a slash (`/`) delimiter. For example,

`=@theothernet/17` or `=@172.16.132/22`

...where the mask is the number of leftmost contiguous significant bits in the corresponding IP address.

When specifying individual IP addresses, use the same `@` notation described above, without a netmask specification. For example:

```
=@172.16.132.14
```

Multiple, individual IP addresses would be specified, for example, as:

```
root=@172.16.132.20:@172.16.134.20
```

A prefixed minus sign (–) denies access to that component of *access_list*. The list is searched sequentially until a match is found that either grants or denies access, or until the end of the list is reached. For example, if host ”terra” is in the ”engineering” netgroup, then

```
rw=-terra:engineering
```

denies access to `terra` but

```
rw=engineering:-terra
```

grants access to `terra`.

オペランド

The following operands are supported:

pathname
: The pathname of the file system to be shared.

使用例

例 1 Define and Publish an NFS Share

The following example shows how to define and publish the `/export/manuals` file system share.

```
# share -F NFS /export/manuals
```

終了ステータス

The following exit values are returned:

`0`
Successful completion.

`>0`
An error occurred.

ファイル

`/etc/dfs/fstypes`
list of system types, NFS by default

`/etc/dfs/sharetab`
system record of shared file systems

`/etc/nfs/nfslogtab`
system record of logged file systems

`/etc/nfs/nfslog.conf`
logging configuration file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

関連項目

`mount(1M)`, `mountd(1M)`, `nfsd(1M)`, `nfslogd(1M)`, `share(1M)`, `unshare(1M)`, `zfs(1M)`, `getnetbyname(3SOCKET)`, `nfslog.conf(4)`, `netgroup(4)`, `attributes(5)`, `nfssec(5)`

注意事項

Creating and publishing an NFS share with the `share` command is permanent until the share is unshared. Publishing NFS shares is managed by the following SMF service:

```
$ svcs | grep share
online         Mar_07   svc:/network/shares:default
```

If the file system being shared is a symbolic link to a valid pathname, the canonical path (the path which the symbolic link follows) are shared. For example, if `/export/foo` is a symbolic link to `/export/bar` (`/export/foo -> /export/bar`), the following `share` command results in `/export/bar` as the shared pathname (and not `/export/foo`).

```
# share -F nfs /export/foo
```

An NFS mount of `server:/export/foo` results in `server:/export/bar` really being mounted.

The `mountd(1M)` process allows the processing of a path name the contains a symbolic link. This allows the processing of paths that are not themselves explicitly shared with `share_nfs`. For example, `/export/foo` might be a symbolic link that refers to `/export/bar` which has been specifically shared. When the client mounts `/export/foo` the `mountd` processing follows the symbolic link and responds with the `/export/bar`. The NFS Version 4 protocol does not

use the `mountd` processing and the client's use of `/export/foo` does not work as it does with NFS Version 2 and Version 3 and the client receives an error when attempting to mount `/export/foo`.

名前

share_smb – make SMB shares available for mounting by remote systems

形式

```
share  -F smb [-a [-o specific-options] [-d description]
     pathname share-name  | [-A]]
zfs set share=name=share-name,path=pathname,[desc="description"],
     prot=smb,[[property=value[,...]] dataset
```

機能説明

The `share` command defines and publishes a SMB share, which makes a local file system available for mounting by remote systems.

You can modify the behavior of SMB shares by setting property values with either the `share` or `zfs set share` command. See the `share(1M)` and `zfs(1M)` man pages.

The `share` command has the following options:

`-F smb`
: Share SMB file sharing protocol.

`-a`
: Publish all defined shares.

`-o` *specific-options*
: Specify *specific-options* in a comma-separated list of keywords and attribute-value-assertions for interpretation by the SMB protocol. By default, a share is published with read-write access to all clients, unless a specific option overrides the default access. *specific-options* can be any combination of the properties supported by a given file system.

`-d` *description*
: Provide a comment that describes the file system to be shared.

`-A`
: Display all defined shares.

Share Properties

The following SMB share properties are supported and can be set by the `zfs` and `share` commands:

`abe=`*boolean*
: Sets the access-based enumeration (ABE) policy for a share. When set to `true`, ABE filtering is enabled on this share and directory entries to which the requesting user has no access will be omitted from directory listings returned to the client. When set to `false` or not defined, ABE filtering will not be performed on this share. This property is not defined by default.

  `false`
  : Disable ABE for this share.

  `true`
  : Enable ABE for this share.

`ad-container`

Specifies the AD container in which to publish shares.

The AD container is specified as a comma-separated list of attribute name-value pairs using the LDAP distinguished name (DN) or relative distinguished name (RDN) format.

The following example uses the `share` command to specify the AD container:

```
$ share -F smb -o name=sh1,abe=true,\
ad-container=cn=sales,ou=mycompany,dc=com /export/home
```

The following example uses the `zfs` command to specify the AD container:

```
$ zfs set share=name=sh1,path=/export/home,prot=smb,\
ad-container=cn=sales,ou=mycompany,dc=com,abe=true rpool/export/home
```

The DN or RDN must be specified in LDAP format using the `cn=`, `ou=`, and `dc=` prefixes:

- `cn` represents the common name
- `ou` represents the organizational unit
- `dc` represents the domain component

`cn=`, `ou=` and `dc=` are attribute types. The attribute type used to describe an object's RDN is called the naming attribute, which, for ADS, includes the following object classes:

- `cn` for the `user` object class
- `ou` for the organizational unit (`OU`) object class
- `dc` for the `domainDns` object class

`catia=`*boolean*

Specifies whether to perform CATIA character substitution. CATIA V4 uses characters in file names that are considered to be invalid by Windows. A CATIA V4 file could be inaccessible to Windows clients if the file name contains any of the characters that are considered illegal in Windows. By default, CATIA character substitution is not performed. See 『Oracle Solaris Administration: SMB and Windows Interoperability』の「Enabling CATIA V4/V5 Character Translations」.

If the `catia` property is set to `true`, the following character substitution is applied to file names.

```
CATIA    CATIA
V4 UNIX  V5 Windows
  "      \250   0x00a8  Dieresis
  *      \244   0x00a4  Currency Sign
  /      \370   0x00f8  Latin Small Letter O with Stroke
  :      \367   0x00f7  Division Sign
  <      \253   0x00ab  Left-Pointing Double Angle Quotation Mark
  >      \273   0x00bb  Right-Pointing Double Angle Quotation Mark
  ?      \277   0x00bf  Inverted Question Mark
  \      \377   0x00ff  Latin Small Letter Y with Dieresis
  |      \246   0x00a6  Broken Bar
```

`csc=`*value*

Sets the client-side caching policy for a share. Client-side caching is a client feature and offline files are managed entirely by the clients.

The following are valid values for the `csc` property:

- `manual` – Clients are permitted to cache files from the specified share for offline use as requested by users. However, automatic file-by-file reintegration is not permitted. `manual` is the default value.
- `auto` – Clients are permitted to automatically cache files from the specified share for offline use and file-by-file reintegration is permitted.
- `vdo` – Clients are permitted to automatically cache files from the specified share for offline use, file-by-file reintegration is permitted, and clients are permitted to work from their local cache even while offline.
- `disabled` – Client-side caching is not permitted for this share.

`dfsroot=`*boolean*

Marks a share as a distributed file system (DFS) root share to distinguish it from a regular share. By default, `dfsroot` is not defined. If `dfsroot` is `false` or not defined, the share is not a DFS root share.

`guestok=`*boolean*

Sets the guest access policy for the share. When set to `true` guest access is allowed on this share. When set to `false` or not defined guest access is not allowed on this share. This property is not defined by default.

An `idmap(1M)` name-based rule can be used to map `guest` to any local user name, such as `guest` or `nobody`. If the local account has a password in `/var/smb/smbpasswd` the guest connection will be authenticated against that password. Any connection made using an account that maps to the local guest account will be treated as a guest connection.

The following name-based rule maps the Windows `Guest` user to the UNIX `guest` user:

```
# idmap add winname:Guest unixuser:guest
```

`none=`*access-list*

Specifies that access is not allowed to any client that matches the access list. The exception is when the access list is an asterisk (`*`), in which case `ro` or `rw` can override `none`.

`ro=`*access-list*

Specifies that sharing is read-only to the clients listed in *access-list*. Overrides the `rw` suboption for the clients specified. See *access-list*.

`rw=`*access-list*

Specifies that sharing is read-write to the clients listed in *access-list*. Overrides the `ro` suboption for the clients specified. See *access-list*.

Access List Argument

The *access-list* argument is either the string "*" to represent all hosts or a colon-separated list whose components may be any number of the following:

hostname
: Specifies the name of a host. *hostname* must be a fully qualified DNS or LDAP name when the host specifies these naming schemes in the `hosts` portion of the `nsswitch.conf` file.

netgroup
: A netgroup contains a number of host names. Any *hostname* in a netgroup must be a fully qualified DNS or LDAP name when the host specifies these naming schemes in the `hosts` portion of the `nsswitch.conf` file.

domainname.suffix
: To use domain membership, the server must use DNS or LDAP to resolve host names to IP addresses. This means that the `hosts` entry of the `/etc/nsswitch.conf` file must specify `dns` or `ldap` before `nis`. You must do this because only DNS and LDAP return the full domain name of the host.

  Other naming services, such as NIS, cannot be used to resolve host names on the server because these naming services do not return domain information. For example, the following shows how NIS, DNS, and LDAP return host name information for the `172.16.45.9` IP address:

NIS	Returns: `myhost`
DNS or LDAP	Returns: `myhost.mydomain.mycompany.com`

  The domain name suffix is distinguished from host names and netgroups by a prefixed dot. For example, `rw=.mydomain.mycompany.com` matches all host names in `mydomain.mycompany.com`.

  The `rw=.` notation uses a single dot to match a host name that has no suffix. This notation matches `mydomain` but not `mydomain.mycompany.com`. This feature can be used to match hosts that are resolved by NIS rather than by DNS and LDAP.

network
: The network or subnet component is preceded by an at-sign character (`@`). It can be either a network name or a dotted address.

  A network name is converted to a dotted address by using `getnetbyname(3SOCKET)`. For example, `=@mynet` is equivalent to `=@172.16` or `=@172.16.0.0`.

  The network prefix assumes an octet-aligned netmask. The netmask is determined from the zeroth octet in the low-order part of the address up to and including the high-order octet. If network prefixes are not byte-aligned, the syntax permits a mask length to be explicitly specified following a slash delimiter (`/`). For example, `=@theothernet/17` or `=@172.16.132/22` where the mask is the number of leftmost contiguous significant bits in the corresponding IP address.

When specifying individual IP addresses, use the same @ notation described previously, but do not use a netmask specification. For example, =@172.16.132.14.

You can use a colon character (:) to separate multiple, individual IP addresses. For example, root=@172.16.132.20:@172.16.134.20.

A prefixed minus sign (–) denies access to that component of *access-list*. The list is searched sequentially until a match is found that either grants or denies access, or until the end of the list is reached. For example, if host "terra" is in the "engineering" netgroup, specifying rw=-terra:engineering denies access to terra. However, specifying rw=engineering:-terra grants access to terra.

使用例

例 1 Setting a Share Property

The following examples use the zfs share and share commands to create and publish an SMB share.

- The following example shows how to use the zfs share command to create and publish an SMB share that also enables guest access:

```
# zfs set share=name=smb_share,path=/mntpnt/dir2,prot=smb,\
guestok=true tank/home
```

- The following example shows how to use the share command to enable guest access on a share:

```
# share -F smb -o guestok=true /mntpnt/dir2
```

例 2 Viewing the Share Properties on a Share

The following examples show how to use the zfs get command and the /etc/dfs/sharetab file to view share properties:

- The zfs get command enables you to view share properties on the tank/home dataset:

```
# zfs get share tank/home
NAME PROPERTY VALUE SOURCE
tank/home share name=smb_share,path=/mntpnt/dir2,guestok=true \
local
```

- The /etc/dfs/sharetab file shows all the active shares on the system. The entry for each share shows the properties set and their values:

```
# cat /etc/dfs/sharetab
/mntpnt/dir2 smb_share smb csc=auto,guestok=true
```

ファイル

/etc/dfs/sharetab
System record of shared file systems

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/smb
Interface Stability	Committed

関連項目

idmap(1M), share(1M), zfs(1M), zfs(1M), getnetbyname(3SOCKET), netgroup(4), attributes(5)

名前

showmount – リモートマウントの表示

形式

`/usr/sbin/showmount [-ade] [hostname]`

機能説明

`showmount` は、ホスト (*host*) からファイルシステムをリモートマウントしているクライアントを表示します。この情報は、*host* 上の `mountd(1M)` サーバーが管理し、クラッシュ後もファイル `/etc/rmtab` 内に保存されます。*host* のデフォルト値は `hostname(1)` が返す値です。

`showmount` コマンドは、NFS バージョン 4 のクライアントの名前を表示しません。

オプション

`-a` すべてのリモートマウント情報を次の形式で表示します。

ホスト名 : ディレクトリ

ホスト名はクライアントの名前で、ディレクトリはマウントされているファイルシステムの元です。

`-d` クライアントがリモートマウントしたディレクトリを表示します。

`-e` 共有ファイルシステムの一覧を表示します。

ファイル

`/etc/rmtab`

属性

次の属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/nfs

関連項目

`hostname(1)`, `mountd(1M)`, `attributes(5)`

『Oracle Solaris 11 システムのインストール』

使用上の留意点

クライアントがクラッシュすると、そのエントリがサーバー上のリモートマウントリストから削除されます。

名前

shutdown – システムのシャットダウンとシステム状態の変更

形式

```
/usr/sbin/shutdown [-y] [-g grace-period] [-r | -i init-state]
      [message]
```

機能説明

shutdown は、マシンの状態を変更するために、スーパーユーザーが実行します。通常は、マルチユーザー状態 (state 2) からほかの状態に移行するために使用されます。

デフォルトでは、shutdown は、コンソール以外はオペレーティングシステムにアクセスできないシステム状態にします。この状態はシングルユーザー状態と呼ばれます。

デーモンをシャットダウンさせたりプロセスを強制終了させたりする前に、shutdown は警告メッセージを出力するとともに、デフォルトでは確認のための最終メッセージも出力します。*message* は、標準の警告メッセージに続いて送信される文字列です:

```
The system will be shut down in . . .
```

この文字列に 2 つ以上の語句を含める場合は、一重引用符 (') または二重引用符 (") で囲む必要があります。

この警告メッセージとユーザーが作成した *message* は、shutdown コマンドが開始されるまでの残り時間が 7200 秒、3600 秒、1800 秒、1200 秒、600 秒、300 秒、120 秒、60 秒、および 30 秒の時点で出力されます。「使用例」を参照してください。

各システム状態の定義は次のとおりです。

state 0　オペレーティングシステムを停止します。

state 1　state 1 は管理状態と呼ばれます。state 1 では、マルチユーザーの操作に必要なファイルシステムがマウントされ、マルチユーザーファイルシステムにアクセスする必要があるログインを使用できます。システムがファームウェアモードから state 1 に移行する時はコンソールだけがアクティブであり、ほかのマルチユーザー (stete 2) サービスは使用できません。マルチユーザー状態から state 1 への移行時にすべてのユーザープロセスが停止されるわけではないことに注意してください。

state s, S　state s (または S) はシングルユーザー状態と呼ばれます。この状態への移行時にはすべてのユーザープロセスが停止されます。シングルユーザー状態では、マルチユーザーログインに必要なファイルシステムのマウントが解除され、システムへのアクセスはコンソールを介してしか行えません。マルチユーザーファイルシステムへアクセスする必要があるログインは使用できません。

state 5　電源を切っても安全なようにマシンを停止します。可能であれば電源を自動的に切断します。rc0 プロシージャがこのために呼び出されます。

state 6　　オペレーティングシステムを停止したあと、/etc/inittab の initdefault エントリに定義されている状態でリブートします。rc6 プロシージャがこのために呼び出されます。

オプション

-y
ユーザーの介入なしでコマンドが実行されるように、確認の問い合わせにあらかじめ応答します。

-g *grace-period*
デフォルトの 60 秒を、スーパーユーザーが変更できるようにします。

-i *init-state*
警告が表示された場合に、init が移行する状態を *init-state* に指定します。デフォルトは、システム状態 s です。

-r
-i6 を指定することと同等です。

使用例

例 1　shutdown の使用

この例では、ホスト foo で shutdown が実行されており、120 秒後にシャットダウンがスケジュールされています。警告メッセージは、最終の確認メッセージの 2 分前、1 分前、および 30 秒前に出力されます。

```
example# shutdown -i S -g 120 "===== disk replacement ====="
Shutdown started.   Tue Jun   7  14:51:40 PDT  1994

Broadcast Message from root (pts/1) on foo Tue Jun  7 14:51:41. . .
The system will be shut down in 2 minutes
===== disk replacement =====
Broadcast Message from root (pts/1) on foo Tue Jun  7 14:52:41. . .
The system will be shut down in 1 minutes
===== disk replacement =====
Broadcast Message from root (pts/1) on foo Tue Jun  7 14:53:41. . .
The system will be shut down in 30 seconds
===== disk replacement =====
Do you want to continue? (y or n):
```

ファイル

/etc/inittab　　init によるプロセスディスパッチを制御します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

boot(1M), halt(1M), init(1M), killall(1M), reboot(1M), ufsdump(1M), init.d(4), inittab(4), nologin(4), attributes(5)

注意事項

システムが S または s 状態へ移行するときに、/etc/nologin ファイル (nologin(4) を参照) が作成されます。その後 state 2 (マルチユーザー状態) へ移行する時点で、このファイルは /etc/rc2.d ディレクトリにあるスクリプトによって削除されます。

名前

slpd – Service Location Protocol Daemon

形式

`/usr/lib/inet/slpd [-f` *configuration-file*`]`

機能説明

The `slpd` daemon provides common server functionality for the Service Location Protocol ("SLP") versions 1 and 2, as defined by IETF in RFC 2165 and RFC 2608. SLP provides a scalable framework for the discovery and selection of network services.

`slpd` provides the following framework services:

Directory Agent	This service automatically caches service advertisements from service agents to provide them to user agents, and makes directory agent advertisements of its services. This service is optional. `slpd` does not provide directory agent service by default. Directory agents are not databases, and they do not need to be maintained.
Service Agent Server	All service agents on the local host register and deregister with this server. This service responds to all requests for services, and forwards registrations to directory agents. By default, `slpd` is a service agent server.
Passive Directory Agent Discovery	This service listens for directory agent advertisements and maintains a table of active directory agents. When a user agent wishes to discover a directory agent, it can simply query `slpd`, obviating the need to perform discovery by means of multicast. By default, `slpd` performs this service.
Proxy Registration	This service can act as a proxy service agent for services that cannot register themselves. `slpd` reads the proxy registration file for information on services it is to proxy. By default, no services are registered by proxy.

All configuration options are available from the configuration file. `slpd` reads its configuration file upon startup.

Stop and start the slpd daemon using `svcadm(1M)`. Use the command `svcadm enable network/slp` to start the slpd daemon. Use the command `svcadm disable network/slp` to stop it.

The file `/etc/inet/slp.conf` must exist before the `slp` service can start the daemon. Only the example file `/etc/inet/slp.conf.example` is present by default. To enable SLP, copy `/etc/inet/slp.conf.example` to `/etc/inet/slp.conf`.

オプション

The following options are supported:

`-f` *configuration-file* Specify an alternate configuration file

使用例

例 1 Stopping the slpd daemon

The following command stops the `slpd` daemon:

```
example# svcadm disable network/slp
```

例 2 Restarting the slpd daemon

The following command restarts the `slpd` daemon:

```
example# svcadm restart network/slp
```

ファイル

`/etc/inet/slp.conf` The default configuration file

`slpd.reg` The proxy registration file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/slp, service/network/slp
CSI	Enabled
Interface Stability	Committed

関連項目

`svcs(1)`, `svcadm(1M)`, `slp_api(3SLP)`, `slp.conf(4)`, `slpd.reg(4)`, `attributes(5)`, `smf(5)`, `slp(7P)`

『Oracle Solaris のシステム管理 (ネットワークサービス)』

Guttman, E., Perkins, C., Veizades, J., and Day, M., RFC 2608, Service Location Protocol, Version 2, The Internet Society, June 1999.

注意事項

The `slpd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/network/slp
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

smattrpop – populate security attribute databases in a name service

形式

```
smattrpop [-c ] [-f] [-m] [-p policy] [-r] -s scope -t scope
     [-v] database
```

機能説明

The smattrpop command updates the auth_attr(4), exec_attr(4), prof_attr(4), and user_attr(4) role-based access control databases in a target NIS, LDAP, or local /etc files name service from the corresponding databases in a source name service or files.

This command processes the table entries from the source database and merges each source entry field into the same field in the corresponding table entry in the target database. If a source entry does not exist in the target database, the entry is created. If the source entry exists in the target database, the fields are merged or replaced according to the command options.

Any errors encountered while updating the target entry are reported to stdout, and the command continues with the next source database entry.

オプション

The following options are supported:

-c

Performs cross-table checking. If you specify this option and a check error occurs, a message identifying the check error is written to stdout.

The target entry values are checked against entries in related databases:

- auths values — Each value must exist as the name of an authorization in the auth_attr(4) database.
- profiles values — Each value must exist as a name of a profile in the prof_attr(4) database.
- roles values — Each value must exist as the name of a role identity in the user_attr(4) database.
- For each exec_attr(4) entry in the source database, the name must exist as the name of a profile in the prof_attr(4) database.

-f

Specifies that the value in each field in the source entry replaces the value in the corresponding field in the target entry, if the source entry field has a non-empty value.

-m

For the auths, profiles, and roles attributes, specifies that the values in each field in the source entry are merged with the values in the corresponding target entry field. If a source value does not exist in the target field, the value is appended to the set of target values. If the target field is empty, the source values replace the target field. The attribute values that merge depend on the database being updated:

- prof_attr(4) — the auths and profiles attribute values are merged.
- user_attr(4) — the auths, profiles, and roles attribute values are merged.
- exec_attr(4) — the uid, gid, euid, and egid values are merged.

`-p` *policy*
: Specifies the value of the policy field in the `exec_attr`(4) database. Valid values are `suser` (standard Solaris superuser) and `tsol` (Trusted Solaris). If you specify this option, only the entries in the source `exec_attr` database with the specified policy are processed. If you omit this option, all entries in the source `exec_attr` database are processed.

`-r`
: Specifies that role identities in the `user_attr`(4) database in the source name service are processed. If you omit this option, only the normal user entries in the `user_attr` source database are processed.

`-s` *scope*
: Specifies the source name service or local file directory for database updates, using the following syntax:

  type`:/`*server*`/`*domain*

  where *type* indicates the type of name service. Valid values for *type* are:

  - `file` — local files
  - `nis` — NIS name service
  - `ldap` — LDAP name service

  server indicates the local host name of the Solaris system on which the `smattrpop` command is executed, and on which both the source and target databases exist.

  domain specifies the management domain name for the name service.

  You can use two special cases of *scope* values:

  - To indicate the databases in the `/etc/security` local system directory, use the scope `file`:/*server*, where *server* is the name of the local system.
  - To load from databases in an arbitrary directory on the Solaris server, use the scope `file`:/*server*/*pathname*, where where *server* is the name of the local system and *pathname* is the fully-qualified directory path name to the database files.

`-t` *scope*
: Specifies the target name service or local file directory for database updates, using the following syntax:

  type`:/`*server*`/`*domain*

  where *type* indicates the type of name service. Valid values for *type* are:

  - `file` — local files
  - `nis` — NIS name service
  - `ldap` — LDAP name service

  server indicates the local host name of the Solaris system on which the `smattrpop` command is executed, and on which both the source and target databases exist.

  domain specifies the management domain name for the name service.

You can use two special cases of *scope* values:

- To indicate the databases in the `/etc/security` local system directory, use the scope `file:/`*server*, where *server* is the name of the local system.
- To update to databases in an arbitrary directory on the Solaris server, use the scope `file:/`*server*/*pathname*, where where *server* is the name of the local system and *pathname* is the fully-qualified directory path name to the database files.

`-v` Specifies that verbose messages are written. A message is written to `stdout` for each entry processed.

オペランド

The following operands are supported:

database Populates one or all databases. You can specify either the name of the database you want to process (for example, `auth_attr`), or `all` to process all databases. If you specify `all`, the databases are processed in the following order:

1. `auth_attr`(4)
2. `prof_attr`(4)
3. `exec_attr`(4)
4. `user_attr`(4)

使用例

例 1 Populating all tables in the NIS name service

The following example merges the values from all four attribute databases in the `/etc/security` directory of the local system into the corresponding tables in the NIS domain, `east.example.com`. The command is executed on the master server, `hoosier`, for the NIS domain and the source files are in the `/etc` and `/etc/security` directories on the NIS master server. No cross-table checking is performed. A summary message indicating the number of entries processed and updated for each table is written to `stdout`.

```
/usr/sadm/bin/smattrpop -s file:/hoosier \
          -t nis:/hoosier/east.example.com all
```

環境変数

See `environ(5)` for a description of the `JAVA_HOME` environment variable, which affects the execution of the `smattrpop` command. If this environment variable is not specified, the `/usr/java` location is used.

終了ステータス

Any errors encountered while updating the target entry are reported to `stdout`. The following exit values are returned:

`0` The specified tables were updated. Individual entries may have encountered checking errors.

`1` A syntax error occurred in the command line.

`2` A fatal error occurred and the tables were not completely processed. Some entries may have been updated before the failure.

ファイル

/etc/security/auth_attr	Authorization description database. See auth_attr(4).
/etc/security/exec_attr	Execution profiles database. See exec_attr(4).
/etc/security/prof_attr	Profile description database. See prof_attr(4).
/etc/user_attr	Extended user attribute database. See user_attr(4).

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWmga

関連項目

auth_attr(4), exec_attr(4), prof_attr(4), user_attr(4), attributes(5), environ(5)

名前

smbadm – SMB ローカルグループおよびユーザーの構成と管理、ドメインメンバーシップの管理、永続的パスワード情報の管理、および各種コマンドの発行

形式

```
smbadm add-key [-u username]

smbadm add-member -m member [[-m member] ...] group

smbadm create-group [-d description] group

smbadm crypt

smbadm delete-group group

smbadm disable-user username

smbadm enable-user username

smbadm get-group [[-p property] ...] group

smbadm join -u username domain

smbadm join -w workgroup

smbadm lookup-server //server

smbadm lookup-user [-u username] name | SID

smbadm print [-u username] //server/share {print_file|-}

smbadm remove-key [-u username]

smbadm remove-member -m member [[-m member] ...] group

smbadm rename-group group new-group

smbadm set-group -p property=value [[-p property=value] ...] group

smbadm show-connections [-t] [-u username] [-c computername | -s sharename]
     server

smbadm show-domains

smbadm show-files [-t] [-u username] server

smbadm show-groups [-m] [-p] [group]

smbadm show-sessions [-t] [-u username] server

smbadm show-shares [-t] [-A | -u username] server
```

機能説明

smbadm コマンドを使用して、SMB ローカルグループの構成、ドメインメンバーシップの管理、永続的パスワード情報の管理、および各種コマンドの発行を行います。また、smbadm コマンドを使用して、個々のローカルユーザーのための SMB パスワードの生成を有効または無効にすることもできます。

SMB ローカルグループは、Windows アカウントが何らかのローカルグループのメンバーである必要がある場合や、Windows スタイルの特権を許可する必要がある場合に使用できます。Solaris ローカルグループは、これらの機能を提供できません。

ローカルグループには、ユーザー定義と組み込みの 2 つのタイプがあります。組み込みローカルグループは、一般的な管理タスクをサポートするための定義済みのローカルグループです。

SMB ローカルグループと Solaris グループの間の正しい識別情報マッピングを提供するには、SMB ローカルグループに、対応する Solaris グループが存在する必要があります。この要件には、次の 2 つの影響があります。まず、グループ名が、Windows と Solaris のグループ名規則の共通部分に従っている必要があります。そのため、SMB ローカルグループ名の最大長は 8 文字であり、小文字と数字のみを含むことができます。2 番目に、Solaris ローカルグループを作成してから、SMB ローカルグループを作成する必要があります。

組み込みグループは標準の Windows グループであり、SMB サービスによって事前に定義されています。組み込みグループは、追加、削除、または名前変更を行うことができず、また SMB ローカルグループの命名規則には従いません。

SMB サーバーの起動時は、次の組み込みグループが使用できます。

Administrators
: グループメンバーはシステムを管理できます。

Backup Operators
: グループメンバーは、ファイルをバックアップおよび復元するためのファイルのアクセス制御をバイパスできます。

Power Users
: グループメンバーはディレクトリを共有できます。

Solaris ローカルユーザーは、認証のため、および SMB リソースへのアクセスを取得するための SMB パスワードを持っている必要があります。このパスワードは、`pam_smb_password` モジュールがシステムの PAM 構成に追加されたときに `passwd(1)` コマンドを使用して作成されます。`pam_smb_passwd(5)` のマニュアルページを参照してください。

`disable-user` サブコマンドと `enable-user` サブコマンドは、指定されたローカルユーザーのための SMB パスワード生成を制御します。無効になっている場合、ユーザーは Solaris SMB サービスに接続できません。デフォルトでは、すべてのローカルユーザーに対して SMB パスワード生成が有効になっています。

無効になっているユーザーを再度有効にするには、`enable-user` サブコマンドを使用したあと、`passwd` コマンドを使用してそのユーザーのパスワードをリセットする必要があります。SMB パスワードを生成するには、`pam_smb_passwd.so.1` モジュールがシステムの PAM 構成に追加されている必要があります。

バックスラッシュ文字のエスケープ

`add-member`、`remove-member`、および `join` (`-u` が指定されている場合) サブコマンドでは、バックスラッシュ文字 (\) は、メンバーまたはユーザー名とドメイン名の間の有効な区切り文字です。バックスラッシュ文字はシェルの特殊文字であるため、クォートする必要があります。たとえば、*domain**username* のように、別の

バックスラッシュ文字でバックスラッシュ文字をエスケープできます。シェルの特殊文字を処理する方法の詳細については、使用しているシェルのマニュアルページを参照してください。

オペランド

`smbadm` コマンドは、次のオペランドを使用します。

domain
参加する既存の Windows ドメインの名前を指定します。

group
SMB ローカルグループの名前を指定します。

username
Windows ユーザーの名前を指定します。*username* は、次のいずれかの形式で指定できます。

*domain**username*[+*password*]
domain/*username*[+*password*]
username@*domain*
username

ここで、*domain* には NetBIOS または DNS ドメイン名を指定できます。

server
ローカルホストの名前または IP アドレスを指定します。

サブコマンド

`smbadm` コマンドには、次のサブコマンドが含まれています。

`add-key` [`-k` *username*]
SMB サーバーのユーザーアカウントに使用される永続的パスワード情報を指定します。この情報を指定すると、Kerberos 以外の構成で、パスワードを要求せずにマウントを行うことができます。Kerberos サイトでは、パスワードを要求せずに Kerberos を自動的に使用するべきです。デフォルトドメインを SMF で使用可能な場合は、ドメインを省略できます。ユーザー名が指定されていない場合は、Solaris のユーザーアカウント名が使用されます。暗号化 (ハッシュ) されたパスワードも使用できます (`crypt` サブコマンドを参照)。 このコマンドは標準入力からパスワードを読み込むこともできます。標準入力が TTY の場合は、プロンプトが表示されます。

また、特定のサーバー用のパスワードを格納するには、ドメイン名の代わりにサーバー名を使用します。

コマンドを実行しているユーザーの永続的パスワードの情報は、`/var/smb/smbfspasswd` にも格納されます。

`add-member -m` *member* [[`-m` *member*] ...] *group*
指定された SMB ローカルグループに指定されたメンバーを追加します。`-m` *member* オプションは、SMB ローカルグループのメンバーの名前を指定します。このメンバー名には、既存のユーザー名とオプションのドメイン名が含まれている必要があります。

メンバー名は、次のどちらかの形式で指定します。

[*domain*\]*username*
[*domain*/]*username*

たとえば、sales\terry や sales/terry は有効なメンバー名です。ここで、sales は Windows ドメイン名であり、terry は sales ドメイン内のユーザーの名前です。

create-group [-d *description*] *group*
指定された名前を持つ SMB ローカルグループを作成します。必要に応じて、-d オプションを使用してグループの説明を指定できます。

crypt
パスワードのハッシュを作成します。このサブコマンドは、パスワードの入力を要求し、ハッシュを標準出力に書き込みます。このハッシュ値は、smbfs マウントおよび各種 smbadm サブコマンドの暗号化パスワードオプションの値として使用することに適しています。

delete-group *group*
指定された SMB ローカルグループを削除します。組み込みグループは削除できません。

disable-user *username*
指定されたローカルユーザーに対する SMB パスワード生成機能を無効にします。無効になっているローカルユーザーは、SMB サービスを使用してシステムにアクセスすることはできません。ローカルユーザーのアカウントが無効になっている場合は、ユーザーアカウントが再度有効になるまで、passwd コマンドを使用してそのユーザーの SMB パスワードを変更することはできません。

enable-user *username*
指定されたローカルユーザーに対する SMB パスワード生成機能を有効にします。パスワード生成機能が再度有効になったあと、ユーザーが SMB サービスに接続する前に、passwd コマンドを使用してそのローカルユーザーの SMB パスワードを生成する必要があります。

pam_smb_passwd モジュールがシステムの PAM 構成に追加されている場合、passwd コマンドは、このユーザーの Solaris パスワードと SMB パスワードの両方を管理します。

get-group [[-p *property=value*] ...] *group*
指定されたグループのプロパティー値を取得します。プロパティーが指定されていない場合は、すべてのプロパティー値が表示されます。

join -u *username domain*
Windows ドメインまたはワークグループに参加します。

SMB サービスのデフォルトモードは、デフォルトのワークグループ名である WORKGROUP を使用するワークグループモードです。

ドメインに参加するには認証されたユーザーアカウントが必要であるため、-u オプションを使用して Windows 管理ユーザーの名前を指定する必要があります。コマンド行でパスワードを指定しない場合、ユーザーはパスワードを入力するように要求されます。このユーザーはドメイン管理者か、またはターゲットドメインに対する管理特権を持つ任意のユーザーであるべきです。

username と *domain* は、次のいずれかの形式で入力できます。

username[+*password*] *domain*
*domain**username*[+*password*]
domain/*username*[+*password*]
username@*domain*

ここで、*domain* には NetBIOS または DNS ドメイン名を指定できます。

システムのマシン信頼アカウントがドメインコントローラ上にすでに存在する場合は、ドメインに参加するときに、認証された任意のユーザーアカウントを使用できます。ただし、マシン信頼アカウントがまだ存在しない場合は、ドメインに参加するにはドメイン上の管理特権を持つアカウントが必要です。

`join -w` *workgroup*
Windows ドメインまたはワークグループに参加します。

`-w` *workgroup* オプションは、`join` サブコマンドを使用しているときに、参加するワークグループの名前を指定します。

`lookup-server` //*server*
指定されたサーバーを IP アドレス、NetBIOS ドメイン、NetBIOS サーバー名に解決します。*server* には、次のいずれかを指定できます。

- NetBIOS ホスト名
- DNS ホスト名
- IP アドレス

`lookup-user` [`-u` *username*] *name* | *SID*
現在のドメインまたは任意の信頼できるドメイン内のアカウントの名前または SID に関する情報を解決します。

`print` [`-u` *username*] //*server*/*share* {*print_file*|-}
指定されたリモートプリンタにファイルを出力します。 *print_file* がハイフン (-) の場合は、標準入力を読み込みます。デフォルトドメインを SMF で使用可能な場合は、ドメインを省略できます。ユーザー名が指定されていない場合は、Solaris のユーザーアカウント名が使用されます。暗号化 (ハッシュ) されたパスワードも使用できます (`crypt` サブコマンドを参照)。 このコマンドは、リダイレクションによりパスワードを取得することもできます。

`remove-key` [`-u` *username*]
コマンドを実行しているユーザーのパスワードを消去します。`/var/smb/smbfspasswd` にある、コマンドを実行しているユーザーのパスワードも消去されます。

名前のユーザー名およびドメイン名の部分は省略可能です。デフォルトドメインを SMF で使用可能な場合は、ドメインを省略できます。*username* が指定されていない場合は、格納されているコマンドを実行しているユーザーのキーがすべて削除されます。

ユーザーのパスワードが特定のサーバーに格納されている場合は、ドメイン名の代わりにサーバー名を指定する必要があります。

`remove-member -m` *member* [[`-m` *member*] …] *group*
指定された SMB ローカルグループから指定されたメンバーを削除します。`-m` *member* オプションは、SMB ローカルグループのメンバーの名前を指定します。このメンバー名には、既存のユーザー名とオプションのドメイン名が含まれている必要があります。

メンバー名は、次のどちらかの形式で指定します。

[*domain*\]*username*
[*domain*/]*username*

たとえば、`sales\terry` や `sales/terry` は有効なメンバー名です。ここで、`sales` は Windows ドメイン名であり、`terry` は `sales` ドメイン内のユーザーの名前です。

`rename-group` *group new-group*
指定された SMB ローカルグループの名前を変更します。このグループはすでに存在する必要があります。組み込みグループは名前を変更できません。

`set-group -p` *property=value* [[`-p` *property=value*] …] *group*
SMB ローカルグループの設定プロパティーを設定します。組み込みグループの説明や特権は変更できません。

`-p` *property=value* オプションは、指定されたグループに対して設定するプロパティーのリストを指定します。

グループ関連のプロパティーは次のとおりです。

`backup=[on|off]`
SMB ローカルグループのメンバーが、ファイルシステムオブジェクトをバックアップするためのファイルのアクセス制御をバイパスできるかどうかを指定します。

`description=`*description-text*
SMB ローカルグループのテキスト説明を指定します。

`restore=[on|off]`
SMB ローカルグループのメンバーが、ファイルシステムオブジェクトを復元するためのファイルのアクセス制御をバイパスできるかどうかを指定します。

take-ownership=[on|off]
SMB ローカルグループのメンバーがファイルシステムオブジェクトの所有権を取得できるかどうかを指定します。

show-domains
現在のワークグループまたはドメインに関する情報を表示します。この情報には通常、ワークグループ名または第一ドメイン名が含まれます。ドメインモードにある場合、この情報には、ドメインコントローラ名や信頼できるドメイン名が含まれます。

出力内の各エントリは、次のいずれかのタグで識別されます。

- [*] 第一ドメイン
- [.] ローカルドメイン
- [-] その他のドメイン
- [+] 選択されたドメインコントローラ

show-groups [-m] [-p] [*group*]
指定された 1 つまたは複数の SMB ローカルグループに関する情報を表示します。グループが指定されていない場合は、すべてのグループの情報が表示されます。-m オプションが指定されている場合は、グループメンバーも表示されます。-p オプションが指定されている場合は、グループ特権も表示されます。

次の一連のサブコマンドは、ローカルまたはリモートサーバー上のユーザーの共有、セッション、接続、および開かれたファイルに関する情報を表示します。

上に示したタイプのデータを表示するには認証されたユーザーアカウントが必要であるため、-u オプションを使用して Windows 管理ユーザーの名前を指定する必要があります。コマンド行でパスワードを指定しない場合、ユーザーはパスワードを入力するように要求されます。このユーザーはドメイン管理者か、またはターゲットドメインに対する管理特権を持つユーザーであるべきです。ユーザー名が指定されていない場合は、Solaris のユーザーアカウント名が使用されます。

username は、「オペランド」で説明されているいずれかの形式で指定できます。

show-connections [-t] [-u *username*] [-c *computername* | -s *sharename*] *server*
サーバー上で作成された SMB ツリー接続に関する情報を表示します。-c オプションは、クライアントへの対象となる接続のコンピュータ名を指定します。-s オプションは、クライアントへの対象となる接続の共有名を指定します。-t オプションは、表示されるコマンドヘッダーを指定します。

show-files [-t] [-u *username*] *server*
サーバー上の SMB を介して開かれたファイルに関する情報を表示します。-t オプションは、表示されるコマンドヘッダーを指定します。

show-sessions [-t] [-u *username*] *server*
サーバー上の SMB ユーザーセッションに関する情報を表示します。-t オプションは、表示されるコマンドヘッダーを指定します。

`show-shares` [`-t`] [`-A` | `-u` *username*] *server*
サーバー上の SMB 共有に関する情報を表示します。`-t` オプションは、表示されるコマンドヘッダーを指定します。`-A` オプションは、匿名ユーザーを指定します。

終了ステータス

次の終了値が返されます。

0　正常終了。

>0　エラーが発生しました。

属性

次の属性の説明については、`attributes(5)` のマニュアルページを参照してください。

属性タイプ	属性値
使用条件	service/file-system/smb
ユーティリティー名とオプション	不確実
ユーティリティーの出力形式	インタフェースではない
`smbadm join`	廃止

関連項目

`passwd(1)`, `groupadd(1M)`, `idmap(1M)`, `idmapd(1M)`, `kclient(1M)`, `mount_smbfs(1M)`, `share(1M)`, `sharectl(1M)`, `smbd(1M)`, `smbstat(1M)`, `smb(4)`, `smbautohome(4)`, `attributes(5)`, `pam_smb_passwd(5)`, `smf(5)`

名前

smbd – SMB サーバーデーモン

形式

/usr/lib/smbd

機能説明

smbd デーモンは、Windows クライアントなどの SMB クライアントからの CIFS/SMB 要求を処理します。このデーモンを実行できるのは、{PRIV_SYS_SMB}、および /var/run ディレクトリに書き込むための十分な特権を持つプロセスだけです。

smbd デーモンは、使用可能なすべてのトランスポートを介して、share(1M) コマンドまたは zfs(1M) set share コマンドを使用して自動的に起動されます。デフォルトでは、smbd は、NBT (NetBIOS-Over-TCP) トランスポートおよび TCP トランスポートを介して起動されます。

smbd が NBT を介して起動された場合は、次のサービスが起動されます。

- NetBIOS ネームサービスが UDP ポート 137 上で起動されます。
- NetBIOS データグラムサービスが UDP ポート 138 上で起動されます。
- NetBIOS セッションサービスが TCP ポート 139 上で起動されます。

smbd デーモンが TCP を介して起動された場合は、SMB サービスが TCP ポート 445 上で起動されます。

同時に実行できる smbd のインスタンスは 1 つだけです。

終了ステータス

次の終了値が返されます。

0　デーモンがクリーンな状態で終了しました。

95　デーモンが致命的エラーで終了しました。

96　デーモンが構成エラーで終了しました。

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/file-system/smb
インタフェースの安定性	不確実

関連項目

ps(1), svcs(1), share(1M), sharectl(1M), smbadm(1M), smbstat(1M), svcadm(1M), zfs(1M), smb(4), smbautohome(4), system(4), attributes(5), smf(5)

注意事項

サービスの有効化、無効化、再起動などの管理操作を smbd サービスで実行するには、svcadm コマンドを使用します。サービスのステータスに対するクエリーを実行するには、svcs コマンドを使用します。

smbd サービスは、サービス管理機能により、サービス識別子 svc:/network/smb/server の下で管理されます。

smbd サービスが無効になっている場合は、その auto_enable プロパティーが false に設定されていないかぎり、share(1M) コマンドまたは zfs set share コマンドによって有効にされます。

名前

smbiod, smbiod-svc – SMB クライアント I/O デーモン

形式

/usr/lib/smbfs/smbiod

/usr/lib/smbfs/smbiod-svc

機能説明

smbiod と smbiod-svc は、SMB クライアントの内部コンポーネントです。ユーザーからアクセスできる外部インタフェースはありません。SMB クライアントの詳細については、mount_smbfs(1M) のマニュアルページを参照してください。

smbiod-svc は、SMF サービス svc:/network/smb/client:default によって実行され、SMB クライアントに必要な smbiod プロセスを起動および管理します。

smbiod は、SMB クライアント接続を要求するユーザーごとの個別のプロセスとして実行される smbiod-svc によって起動されます。これらの smbiod プロセスは、所有しているユーザーが SMB クライアント接続を使用しなくなるまで実行を継続します。

ファイル

/usr/lib/smbfs/smbiod
: SMB I/O デーモン。

/usr/lib/smbfs/smbiod-svc
: SMF サービスプログラム。

/var/run/smbiod/.svc
: smbiod-svc が door をインスタンス化する対象のファイル。

/var/run/smbiod/$*UID*
: ユーザーの smbiod プロセスが door をインスタンス化する対象のファイル。$*UID* は、所有者のユーザー ID の数値表現です。

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/smb
インタフェースの安定性	不確実

関連項目

ps(1), mount_smbfs(1M), attributes(5), smbfs(7FS)

名前 smbios – System Management BIOS イメージの内容の表示

形式 `smbios [-BeOsx] [-i id] [-t type] [-w file] [file]`

機能説明

`smbios` ユーティリティーは、現在のシステムによってエクスポートされたか、またはファイルに格納されている SMBIOS (System Management BIOS) イメージの内容を表示します。SMBIOS は、低レベルのシステムソフトウェアがハードウェア構成情報をより高レベルのシステム管理ソフトウェアにエクスポートするための、業界標準のメカニズムです。SMBIOS のデータ形式自体は、DMTF (Distributed Management Task Force) によって定義されます。SMBIOS の詳細、および SMBIOS の仕様や実装ガイドラインのコピーを入手する方法については、http://www.dmtf.org を参照してください。

SMBIOS イメージは構造体のテーブルで構成されており、各テーブルにはシステムソフトウェアまたはハードウェアの構成の一部が記述されています。デフォルトでは、`smbios` は、現在の SMBIOS イメージの内容全体を表示します。`-s` オプションが指定されている場合、`smbios` は、このイメージ内に存在する構造体の要約を表示します。`-w` オプションが指定されている場合、`smbios` は、SMBIOS イメージのコピーを指定されたファイルに書き込みます。その後、生成されたファイルに `smbios` を適用すると、その内容を表示できます。

`smbios` は、各構造体とその内容を、人間が読める形式で表示しようとします。`smbios` が構造体のタイプまたは内容を認識できない場合は、その構造体の 16 進数の生データが表示されます。

オプション

次のオプションがサポートされています。

`-B` 壊れた BIOS に対するヘッダー検証を無効にします。

デフォルトでは、`smbios` は、アンカー文字列、ヘッダーチェックサム、およびバージョン番号を確認することによって SMBIOS ヘッダーを検証しようとします。このオプションは、BIOS に非準拠のヘッダーが含まれている場合に必要になることがあります。

`-e` SMBIOS 構造体テーブルの内容ではなく、SMBIOS エントリポイントの内容を表示します。

`-i` *id* 整数の ID で指定された特定の構造体のみを表示します。

`-O` 廃止された構造体タイプを表示します。

デフォルトでは、`smbios` は、そのタイプが DMTF SMBIOS 仕様で廃止としてマークされた構造体の出力を省略します。

`-s` 選択された各構造体の内容の代わりに、構造体の識別子とタイプの要約リストのみを表示します。

`-t` *type* DMTF SMBIOS 仕様で定義されている指定の整数型にタイプが一致する構造体のみを表示します。

-w *file* SMBIOS イメージのコピーを指定されたファイルに書き込み、終了します。

SMBIOS エントリポイントがファイルの先頭に書き込まれ、その構造体テーブルのアドレスが構造体テーブルのファイルオフセットに設定されます。また、新しいエントリポイントチェックサムが計算されます。

-x 選択された構造体について、人間が読める形式の出力に加えて、16 進数の生データを表示します。

デフォルトでは、16 進数データは、選択された構造体について人間が読める形式の出力を smbios が表示できない場合にのみ表示されます。

オペランド

次のオペランドがサポートされています。

file 現在のシステムの SMBIOS イメージの代わりに表示する代替 SMBIOS イメージを指定します。

終了ステータス

次の終了値が返されます。

0 正常終了。SMBIOS イメージ内のすべての構造体が検査され、問題がありませんでした。

1 指定されたファイルまたはデバイスを開けなかったり、イメージが壊れていたりするなど、致命的なエラーが発生しました。

2 無効なコマンド行オプションが指定されました。

ファイル

/dev/smbios カーネルの SMBIOS イメージデバイス。このデバイス特殊ファイルは、現在のシステム SMBIOS イメージのスナップショットをエクスポートするために使用されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照。

コマンド行オプションは「確実」です。人間が読める形式の出力は「不確実」です。

関連項目

prtdiag(1M), attributes(5), smbios(7D)

『System Management BIOS Reference Specification』(http://www.dmtf.org を参照)

注意事項

System Management BIOS イメージの実装は、システムベンダーおよび BIOS ベンダーの判断で行われます。すべてのシステムが SMBIOS をエクスポートするわけではありません。SMBIOS 構造体の内容はシステムベンダーおよび BIOS ベンダーによって大きく異なり、その仕様に含まれているガイドラインに準拠していないことがよくあります。構造体フィールドによっては、BIOS による入力が行われなかったり、準拠しない値が入力されたりする場合があります。

名前

smbstat – Solaris SMB ファイルサーバーの統計情報の表示

形式

`smbstat [-r [-n [-a | -z]] [-t] [-u] [-c] interval`

機能説明

`smbstat` コマンドは、`smbd(1M)` サーバーの統計情報を表示します。`smbstat` には、後述するいくつかのオプションと、1 つのオペランド *interval* があります。*interval* が指定されている場合、最初の表示では、サーバーが起動されてからこのコマンドが入力されるまでの統計情報が取得されます。以降の表示では、最後の *interval* の統計情報が取得されます。

デフォルトでは、`smbstat` コマンドはすべての統計情報を表示します。

オプション

`smbstat` コマンドに含まれているオプションは、次のとおりです。

`-c`
カウンタを表示します。

`-r`
要求の統計情報を表示します。`-r` は次のオプションと組み合わせることができます。

`-a`
有効かどうかにかかわらず、すべてのタイプのサーバー要求の統計情報を表示します。サーバー要求には 256 のタイプがあります。`-a` オプションと `-z` オプションを同時に指定することはできません。

`-n`
アルファベット順に表示します。

`-z`
実際に受信された要求の統計情報を表示します。`-a` オプションと `-z` オプションを同時に指定することはできません。

`-t`
SMB サーバーのスループットを表示します。

`-u`
SMB サーバーの使用率を表示します。

拡張機能説明

`smbstat` コマンドは、下に示す見出しを表示します。あるコマンドで表示される見出しは、どのオプションが指定されたかによって異なります。これらの見出しには、統計情報のカテゴリに対応して、カウンタ、スループット、使用率、および (サーバー) 要求の 4 つのカテゴリがあります。

カウンタ

`nbt`
SMB NBT (NetBIOS-over-TCP) セッションの数。

`tcp`
SMB TCP セッションの数。

users
ログインしているユーザーの数。

trees
接続されているツリーの数。

files
開かれているファイルの数。

pipes
開かれているパイプの数。

スループット

rbytes/s
1 秒あたりに受信されたバイト数。

tbytes/s
1 秒あたりに送信されたバイト数。

reqs/s
1 秒あたりに処理された要求の数。

reads/s
1 秒あたりの読み取り要求の数。これは、SMB_COM_READ、SMB_COM_LOCK_AND_READ、SMB_COM_READ_RAW、および SMB_COM_READ_ANDX の各要求を合わせた数になります。

writes/s
1 秒あたりの書き込み要求の数。これは、SMB_COM_WRITE、SMB_COM_WRITE_AND_UNLOCK、SMB_COM_WRITE_RAW、および SMB_COM_WRITE_AND_CLOSE の各要求を合わせた数になります。

使用率

wcnt
受信されたものの、SMB ワーカースレッドによって実行されるまで待機している要求の平均数。

rcnt
SMB ワーカースレッドによって同時に実行されている要求の平均数。

wtime
要求が SMB ワーカースレッドによって実行されるまで待機している平均時間。

rtime
要求の平均実行時間。

w%
少なくとも 1 つの要求が待機していた時間の割合 (パーセント)。

r%
少なくとも 1 つの要求が実行されていた時間の割合 (パーセント)。

u%
: SMB サーバーの使用率の割合 (パーセント)。この数値は、rcnt / (*max_worker_threads*) として定義されます。

sat
: サーバーが以前に飽和状態になっていたかどうかを示すフラグ。飽和は、u% == 100% として定義されます。

usr%
: 1 つまたは複数のプロセッサがユーザー空間内で費やした時間の割合 (パーセント)。

sys%
: 1 つまたは複数のプロセッサがカーネル空間内で費やした時間の割合 (パーセント)。

idle%
: 1 つまたは複数のプロセッサがアイドル状態であった時間の割合 (パーセント)。

要求

要求のタイプごとに、次の見出しが表示されます。

code
: 要求のコード (16 進数)。

%
: 特定のタイプの要求の割合 (パーセント)。

count
: 受信された要求の数。

rbytes/s
: 1 秒あたりに受信されたバイト数。

tbytes/s
: 1 秒あたりに送信されたバイト数。

req/s
: 1 秒あたりに処理された要求の数。

rt-mean
: 秒数での平均応答時間。

rt-stddev
: 応答時間の標準偏差。

使用例

例 1 オプションの組み合わせ

次のコマンドでは、-c、-t、および -u オプションを組み合わせています。

```
% smbstat -ctu
nbt   tcp users trees files pipes
0     1     1     2    20     0
```

例 1 オプションの組み合わせ (続き)

```
rbytes/s   tbytes/s    reqs/s     reads/s   writes/s
1.036e+02  1.298e+00  0.000e+00  0.000e+00  0.000e+00

wcnt       rcnt       wtime      rtime       w%   r%  u% sat usr% sys% idle%
4.317e-01  7.410e+00  2.461e-05  4.224e-04   31  100  0  no  0    76   24
```

終了ステータス

次の終了値が返されます。

0 正常終了。

>0 エラーが発生しました。

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/file-system/smb
インタフェースの安定性	不確実
ユーティリティーの出力形式	インタフェースではない

関連項目

sharectl(1M), smbadm(1M), smbd(1M), attributes(5)

名前

smrsh – restricted shell for sendmail

形式

`smrsh -c` *command*

機能説明

The `smrsh` program is intended as a replacement for the `sh` command in the `prog` mailer in `sendmail(1M)` configuration files. The `smrsh` program sharply limits commands that can be run using the `|program` syntax of `sendmail`. This improves overall system security. `smrsh` limits the set of programs that a programmer can execute, even if `sendmail` runs a program without going through an `alias` or `forward` file.

Briefly, `smrsh` limits programs to be in the directory `/var/adm/sm.bin`, allowing system administrators to choose the set of acceptable commands. It also rejects any commands with the characters: `,`, `<`, `>`, `|`, `;`, `&`, `$`, `\r` (RETURN), or `\n` (NEWLINE) on the command line to prevent end run attacks.

Initial pathnames on programs are stripped, so forwarding to `/usr/ucb/vacation`, `/usr/bin/vacation`, `/home/server/mydir/bin/vacation`, and `vacation` all actually forward to`/var/adm/sm.bin/vacation`.

System administrators should be conservative about populating `/var/adm/sm.bin`. Reasonable additions are utilities such as `vacation(1)` and `procmail`. Never include any shell or shell-like program (for example, `perl`) in the `sm.bin` directory. This does not restrict the use of `shell` or `perl` scrips in the `sm.bin` directory (using the `#!` syntax); it simply disallows the execution of arbitrary programs.

オプション

The following options are supported:

`-c` *command* Where *command* is a valid command, executes *command*.

ファイル

`/var/adm/sm.bin` directory for restricted programs

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`sendmail(1M)`, , `attributes(5)`

名前

smtnrhdb – manage entries in the tnrhdb database

形式

`/usr/sadm/bin/smtnrhdb` *subcommand* [*auth_args*] -- *subcommand_args*]

機能説明

The `smtnrhdb` command adds, modifies, deletes, and lists entries in the `tnrhdb` database.

The `tnrhdb` database specifies which remote-host template to use for each host, including the local host, in the distributed system. If a host's IP address cannot be matched to some entry in the `tnrhdb` database, communication with the host is not permitted.

Valid Host Addresses and Wildcards

The trusted network software uses a network “longest prefix of matching bits” mechanism when looking for a host. The software looks first for the IP address of the host. If the software does not find this address, then the software falls back to searching for an IP address with the longest prefix of a matching bit pattern, and so on.

注 – The actual numeric value of the subnet address or other subnetting information on the system (for example, from the `netmasks(4)` file) are not considered by this mechanism.

Using the “longest prefix of matching bits” mechanism, an IPv4 address of 0.0.0.0 is a wildcard address with a prefix length of 0 and hence matches any IPv4 address. For more information about prefi x lengths in IPv4 and IPv6 addresses, see 『Oracle Solaris の管理: IP サービス』.

The `smtnrhdb` command accepts a hostname, IP address, and wildcard address with as optional prefix as valid addresses. See *subcommand_args*, below, for the format of valid addresses.

サブコマンド

`smtnrhdb` *subcommand*s are:

`add`
: Adds a new entry to the `tnrhdb` database. To add an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

`delete`
: Deletes an entry from the `tnrhdb` database. To delete an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

`list`
: Lists all entries in the `tnrhdb` database. To list an entry, the administrator must have the `solaris.network.host.read` and `solaris.network.security.read` authorizations.

`modify`
: Modifies an entry in the `tnrhdb` database. To modify an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

オプション

The `smtnrhdb` authentication arguments, *auth_args*, are derived from the `smc` arg set. These arguments are the same regardless of which subcommand you use.

The subcommand-specific options, *subcommand_args*, must be preceded by the `--` option.

auth_args

The valid *auth_args* are `-D`, `-H`, `-l`, `-p`, `-r`, and `-u`; they are all optional. If no *auth_args* are specified, certain defaults will be assumed and the user might be prompted for additional information, such as a password for authentication purposes. These letter options can also be specified by their equivalent option words preceded by a double dash. For example, you can use either `-D` or `--domain`.

`-D` | `--domain` *domain*
: Specifies the default domain that you want to manage. The syntax of *domain=type:/host_name/domain_name*, where *type* is `dns`, `ldap`, or `file`; *host_name* is the name of the server; and *domain_name* is the name of the domain you want to manage.

  If you do not specify this option, the system assumes the `file` default domain on whatever server you choose to manage, meaning that changes are local to the server. Toolboxes can change the domain on a tool-by-tool basis; this option specifies the domain for all other tools.

`-H` | `--hostname` *host_name:port*
: Specifies the *host_name* and *port* to which you want to connect. If you do not specify a *port*, the system connects to the default port, `898`. If you do not specify *host_name:port*, the system connects to the local host on port `898`.

`-l` | `--rolepassword` *role_password*
: Specifies the password for the *role_name*. If you specify a *role_name* but do not specify a *role_password*, the system prompts you to supply a *role_password*. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-p` | `--password` *password*
: Specifies the password for the *user_name*. If you do not specify a password, the system prompts you for one. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-r` | `--rolename` *role_name*
: Specifies a role name for authentication. If you do not specify this option, no role is assumed.

`-u` | `--username` *user_name*
: Specifies the user name for authentication. If you do not specify this option, the user identity running the console process is assumed.

`--`
: This option is required and must always follow the preceding options. If you do not enter the preceding options, you must still enter the `--` option.

subcommand_args

Note: Descriptions and other arg options that contain white spaces must be enclosed in double quotes.

`-h`
Displays the command's usage statement.

`-H` *hostname*
Specifies the name of the host. For the list subcommand, the hostname argument is not specified. This is not required if the ipaddress subcommand argument is specified.

`-i` *ipaddress*
Specifies the IP address of the host. This is not required if the hostname subcommand argument is specified. This option is not valid with the `-w` option.

`-n` *templatename*
Specifies the name of an existing template.

`-p` *prefixlen*
Specifies the prefix length (in bits) of a wildcard representation of the IP address. The prefix is the left-most portion of the IP address. This option is valid only with the `-w` option. For example, when the value of `-w` *ipaddress-wildcard* is 192.168.0.0, a *prefixlen* value of 24 indicates that the wildcard matches all addresses on the 192.168.0 network. With a *prefixlen* of 32, the wildcard 192.168.0.0 matches all addresses on the 192.168.0.0 network.

`-w` *ipaddress-wildcard*
Specifies the IP address of the subnet using a wildcard.

- One of the following sets of arguments must be specified for subcommand `add`:

  -H *hostname* -n *templatename* |
  -i *ipaddress* -n *templatename* |
  -w *ipaddress-wildcard* -n *templatename* [-p *prefixlen*] |
  -h

- One of the following sets of arguments must be specified for subcommand `modify`:

  -H *hostname* -n *templatename* |
  -i *ipaddress* -n *templatename* |
  -w *ipaddress-wildcard* -n *templatename* [-p *prefixlen*] |
  -h

- One of the following sets of arguments must be specified for subcommand `delete`:

  -H *hostname* |
  -i *ipaddress* |
  -w *ipaddress-wildcard* [-p *prefixlen*] |
  -h

- The subcommand `list` takes the following argument:

  -h

使用例

例 1 Specifying the Template Name for a Wildcard IP Address

The admin role specifies the template name, `cipso_lan`, for a series of hosts that use the IP address wildcard `192.168.113.0` on the local file system. Since no authorization arguments were specified, the administrator connects to port 898 of the local host on the local server with the file domain type, which are the defaults. The administrator is prompted for the admin password.

```
$ usr/sadm/bin/smtnrhdb add -- -w 192.168.113.0 -n cipso_lan
```

例 2 Deleting an Entry in the tnrhdb Database

The admin role connects to port 898 (which happens to be the default) of the LDAP server and deletes a host entry from the database by specifying its IP address, `192.168.113.8`. Since the domain was not specified, the file domain type and local server are used by default. The administrator is prompted for the admin password.

```
# /usr/sadm/bin/smtnrhdb delete \
-D ldap:/example.domain -i 192.168.113.8
```

例 3 Adding a Subnet to the tnrhdb Database

The following command adds all the addresses on the 192.168.55.0 subnet, from 192.168.55.1 to 192.168.55.255, to the `tnrhdb` database:

```
# /usr/sadm/bin/smtnrhdb add \
-D file:/machine1.ExampleCo.COM/machine1.ExampleCo.COM \
 -- -w 192.168.55.0 -n cipso
Authenticating as user: root
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.exampleco.admin.hostmgr.cli.smtnrhdb.HostMgrTnrhdbCli
from machine1.ExampleCo.COM
Login to machine1.ExampleCo.COM as user root was successful.
Download of com.exampleco.admin.hostmgr.cli.smtnrhdb.HostMgrTnrhdbCli
from machine1.ExampleCo.COM
was successful.
```

例 4 Adding Subnet 192.168.0 to the tnrhdb Database

The following command adds all the addresses on the 192.168.0 subnet, from 192.168.0.1 to 192.168.0.255 to the `tnrhdb` database. The prefix, 24, indicates that the first 24 bits (192.168.0) are fixed. Only the final zero is a wildcard.

```
# /usr/sadm/bin/smtnrhdb add \
-D file:/machine1.ExampleCo.COM/machine1.ExampleCo.COM \
 -- -w 192.168.0.0 -p 24 -n cipso

Login to machine1.ExampleCo.COM as user root was successful.
Download of com.exampleco.admin.hostmgr.cli.smtnrhdb.HostMgrTnrhdbCli
from machine1.ExampleCo.COM was successful.
```

終了ステータス

The following exit values are returned:

`0`
Successful completion.

`1`
Invalid command syntax. A usage message displays.

`2`
An error occurred while executing the command. An error message displays.

ファイル

The following files are used by the `smtnrhdb` command:

`/etc/security/tsol/tnrhdb`
Trusted network remote-host database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWmgts
Interface Stability	Committed

関連項目

`netmasks(4)`, `attributes(5)`

『Oracle Solaris の管理: セキュリティーサービス』

注意事項

The functionality described on this manual page is available only if the system is configured with Trusted Extensions.

名前

smtnrhtp – manage entries in the trusted network template database

形式

`/usr/sadm/bin/smtnrhtp` *subcommand* [*auth_args*] -- [*subcommand_args*]

機能説明

The `smtnrhtp` command adds, modifies, deletes, and lists entries in the `tnrhtp` database.

`smtnrhtp` *subcommand*s are:

`add` Adds a new entry to the `tnrhtp` database. To add an entry, the administrator must have the `solaris.network.security.read` and `solaris.network.security.write` authorizations.

`modify` Modifies an entry in the `tnrhtp` database. To modify an entry, the administrator must have the `solaris.network.security.read` and `solaris.network.security.write` authorizations.

`delete` Deletes an entry from `tnrhtp` database. To delete an entry, the administrator must have the `solaris.network.security.read` and `solaris.network.security.write` authorizations.

`list` Lists entries in the `tnrhtp` database. To list an entry, the administrator must have the `solaris.network.security.read` authorizations.

オプション

The subcommand-specific options, *subcommand_args*, must be *preceded* by the `--` option.

auth_args

The valid *auth_args* are `-D`, `-H`, `-l`, `-p`, `-r`, and `-u`; they are all optional. If no *auth_args* are specified, certain defaults will be assumed and the user might be prompted for additional information, such as a password for authentication purposes. These letter options can also be specified by their equivalent option words preceded by a double dash. For example, you can use either `-D` or `--domain`.

`-D` | `--domain` *domain*
Specifies the default domain that you want to manage. The syntax of *domain=type:/host_name/domain_name*, where *type* is `dns`, `ldap`, or `file`; *host_name* is the name of the server; and *domain_name* is the name of the domain you want to manage.

If you do not specify this option, the system assumes the `file` default domain on whatever server you choose to manage, meaning that changes are local to the server. Toolboxes can change the domain on a tool-by-tool basis; this option specifies the domain for all other tools.

`-H` | `--hostname` *host_name:port*
Specifies the *host_name* and *port* to which you want to connect. If you do not specify a *port*, the system connects to the default port, `898`. If you do not specify *host_name:port*, the system connects to the local host on port `898`.

`-l` | `--rolepassword` *role_password*
Specifies the password for the *role_name*. If you specify a *role_name* but do not specify a *role_password*, the system prompts you to supply a *role_password*. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-p` | `--password` *password*
Specifies the password for the *user_name*. If you do not specify a password, the system prompts you for one. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-r` | `--rolename` *role_name*
Specifies a role name for authentication. If you do not specify this option, no role is assumed.

`-u` | `--username` *user_name*
Specifies the user name for authentication. If you do not specify this option, the user identity running the console process is assumed.

`--`
This option is required and must always follow the preceding options. If you do not enter the preceding options, you must still enter the `--` option.

subcommand_args

Descriptions and other argument options that contain white spaces must be enclosed in double quotes.

`-h`	Displays the command's usage statement.
`-n` *templatename*	Specifies the name of the template.
`-t` *hosttype*	Specifies the host type of the new host. Valid values are `unlabeled` and `cipso`. The `cipso` host type is for hosts that use CIPSO (Common IP Security Options - Tag Type 1 only) to label packets.
`-x doi=`*doi-value*	Specifies the DOI value (the domain of interpretation). In the case of the `unlabeled` host type, this is the domain of interpretation for the `def_label`. The domain of interpretation defines the set of rules for translating between the external or internal representation of the security attributes and their network representation. When systems that are configured with Trusted Extensions software have the same `doi`, they share that set of rules. In the case of the `unlabeled` host type, these systems also share the same interpretation for the default attributes that are assigned to the unlabeled templates that have that same `doi`.

`-x max=`*maximum-label*	Specifies the maximum label. Together with `min`, this value specifies the label accreditation range for the remote hosts that use this template. Values can be a hex value or string (such as `admin_high`).
`-x min=`*minimum-label*	Specifies the minimum label. Together with `max`, this value specifies the label accreditation range for the remote hosts that use this template. For gateway systems, `min` and `max` define the default range for forwarding labeled packets. The label range for routes is typically set by using a `route(1M)` subcommand with the `-secattr` option. When the label range for routes is not specified, the `min` to `max` range in the security template is used. Values can be a hex value or string (such as `admin_low`).
`-x label=`*default-label*	Specifies the default label to be applied to incoming data from remote hosts that do not support these attributes. This option does not apply if *hosttype* is `cipso`. Values can be a hex value or string (such as `admin_low`).
`-x slset=`*l1*,*l2*,*l3*,*l4*	Specifies a set of sensitivity labels. For gateway systems, the labels in `slset` are used for forwarding labeled packets. `slset` is optional. You can specify up to four label values, separated by commas. Values can be a hex value or string (such as `admin_low`).

- One of the following sets of arguments must be specified for subcommand `add`:

  `-n` *template name* (

  `-t cipso` [`-x doi=`*doi-value* `-x min=`*minimum-label* `-x max=`*maximum-label* `-x slset=`*l1*,*l2*,*l3*,*l4*] |

  `-t` unlabeled [`-x doi=`*doi-value* `-x min=`*minimum-label* `-x max=`*maximum-label* `-x label=`*default-label* `-x slset=`*l1*,*l2*,*l3*,*l4*] |

  `-h`

)

- One of the following sets of arguments must be specified for subcommand `modify`:

  `-n` *template name* (

  `-t cipso` [`-x doi=`*doi-value* `-x min=`*minimum-label* `-x max=`*maximum-label* `-x slset=`*l1*,*l2*,*l3*,*l4*] |

  `-t` unlabeled [`-x doi=`*doi-value* `-x min=`*minimum-label* `-x max=`*maximum-label* `-x label=`*default-label*`-x slset=`*l1*,*l2*,*l3*,*l4*] |

  `-h`

)

If the host type is changed, all options for the new host type must be specified.

- One of the following sets of arguments must be specified for subcommand `delete`:

  `-n` *templatename* |
  `-h`

- The following argument can be specified for subcommand `list`:

  `-n` *templatename* |
  `-h`

使用例

例 1 Adding a New Entry to the Network Template Database

The admin role connects to port `898` of the LDAP server and creates the `unlabeled_ntk` entry in the `tnrhtp` database. The new template is assigned a host type of `unlabeled`, a domain of interpretation of `1`, minimum label of `public`, maximum label of `restricted`, and a default label of `needtoknow`. The administrator is prompted for the admin password.

```
$ /usr/sadm/bin/smtnrhtp \
add -D ldap:directoryname -H servername:898 -- \
-n unlabeled_ntk -t unlabeled -x DOI=1 \
-x min=public -x max=restricted -x label="need to know"
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Invalid command syntax. A usage message displays.

`2` An error occurred while executing the command. An error message displays.

ファイル

The following files are used by the `smtnrhtp` command:

`/etc/security/tsol/tnrhtp` Trusted network remote-host templates.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWmgts
Interface Stability	Committed

関連項目

`attributes(5)`

注意事項

The functionality described on this manual page is available only if the system is configured with Trusted Extensions.

警告

Changing a template while the network is up can change the security view of an undetermined number of hosts.

Allowing unlabeled hosts onto a Solaris Trusted Extensions network is a security risk. To avoid compromising the rest of your network, such hosts must be *trusted* in the sense that the administrator is certain that these unlabeled hosts will not be used to compromise the distributed system. These hosts should also be physically protected to restrict access to authorized individuals. If you cannot guarantee that an unlabeled host is physically secure from tampering, it and similar hosts should be isolated on a separate branch of the network.

If the security template is modified while the network is up, the changes do not take effect immediately unless `tnctl(1M)` is used to update the template entries. Otherwise, the changes take effect when next polled by the trusted network daemon, `tnd(1M)`. Administrators are allowed to add new templates and modify attributes of existing templates while the network is up.

名前

smtnzonecfg – manage entries in the zone configuration database for Trusted Extensions networking

形式

`/usr/sadm/bin/smtnzonecfg` *subcommand* [*auth_args*] `--` [*subcommand_args*]

機能説明

The `smtnzonecfg` command adds, modifies, deletes, and lists entries in the `tnzonecfg` database.

`smtnzonecfg` *subcommand*s are:

`add` Adds a new entry to the `tnzonecfg` database. To add an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

`modify` Modifies an entry in the `tnzonecfg` database. To modify an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

`delete` Deletes an entry from the `tnzonecfg` database. To delete an entry, the administrator must have the `solaris.network.host.write` and `solaris.network.security.write` authorizations.

`list` Lists entries in the `tnzonecfg` database. To list an entry, the administrator must have the `solaris.network.host.read` and `solaris.network.security.read` authorizations.

オプション

The `smtnzonecfg` authentication arguments, *auth_args*, are derived from the `smc` argument set and are the same regardless of which subcommand you use.

The subcommand-specific options, *subcommand_args*, must be *preceded* by the `--` option.

auth_args

The valid *auth_args* are `-D`, `-H`, `-l`, `-p`, `-r`, and `-u`; they are all optional. If no *auth_args* are specified, certain defaults will be assumed and the user can be prompted for additional information, such as a password for authentication purposes. These letter options can also be specified by their equivalent option words preceded by a double dash. For example, you can use either `-D` or `--domain`.

`-D` | `--domain` *domain*

Specifies the default domain that you want to manage. The syntax of *domain=type:/host_name/domain_name*, where *type* is `dns`, `ldap`, or `file`; *host_name* is the name of the server; and *domain_name* is the name of the domain you want to manage.

If you do not specify this option, the system assumes the `file` default domain on whatever server you choose to manage, meaning that changes are local to the server. Toolboxes can change the domain on a tool-by-tool basis. This option specifies the domain for all other tools.

`-H` | `--hostname` *host_name:port*
Specifies the *host_name* and *port* to which you want to connect. If you do not specify a *port*, the system connects to the default port, `898`. If you do not specify *host_name:port*, the system connects to the local host on port `898`.

`-l` | `--rolepassword` *role_password*
Specifies the password for the *role_name*. If you specify a *role_name* but do not specify a *role_password*, the system prompts you to supply a *role_password*. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-p` | `--password` *password*
Specifies the password for the *user_name*. If you do not specify a password, the system prompts you for one. Passwords specified on the command line can be seen by any user on the system, hence this option is considered insecure.

`-r` | `--rolename` *role_name*
Specifies a role name for authentication. If you do not specify this option, no role is assumed.

`-u` | `--username` *user_name*
Specifies the user name for authentication. If you do not specify this option, the user identity running the console process is assumed.

`--`
This option is required and must always follow the preceding options. If you do not enter the preceding options, you must still enter the `--` option.

subcommand_args

Descriptions and other argument options that contain white spaces must be enclosed in double quotes.

`-h`
Displays the command's usage statement.

`-n` *zonename*
Specifies the zone name for the entry. This name is used when the zone is configured. See `zonecfg(1M)`, under the `-z zonename` option, for the constraints on zone names. The specified zone name must be one of the configured zones on the system. The following command returns a list of configured zones:

```
/usr/sbin/zoneadm list -c
```

`-l` *label*
Specifies the label for the zone. This field is used to label the zone when the zone is booted. Each zone must have a unique label.

`-x policymatch=0|1`
Specifies the policy match level for non-transport traffic. Only values of `0` (match the label) or `1` (be within the label range of the zone) are accepted.

ICMP packets that are received on the global zone IP address are accepted based on the label range of the global zone's security template if the global zone's *policymatch* field is set to 1. When this field is set to 0 for a zone, the zone will not respond to an ICMP echo request from a host with a different label.

This subcommand argument is optional. If not specified, it will have a default value of 0.

`-x mlpzone=“”|`*port/protocol*
Specifies the multilevel port configuration entry for zone-specific IP addresses. Multiple *port/protocol* combinations are separated by a semi-colon. The empty string can be specified to remove all existing MLP zone values. This subcommand argument is optional.

An MLP is used to provide multilevel service in the global zone as well as in non-global zones. As an example of how a non-global zone can use an MLP, consider setting up two labeled zones, `internal` and `public`. The `internal` zone can access company networks; the `public` zone can access public internet but not the company's internal networks. For safe browsing, when a user in the `internal` zone wants to browse the Internet, the `internal` zone browser forwards the URL to the `public` zone, and the web content is then displayed in a `public` zone web browser. That way, if the download in `public` zone compromises the web browser, it cannot affect the company's internal network. To set this up, TCP port `8080` in the `public` zone is an MLP (`8080/tcp`), and the security template for the `public` zone has a label range from `PUBLIC` to `INTERNAL`.

`-x mlpshared=“”|`*port/protocol*
Specifies the multilevel port configuration entry for shared IP addresses. Multiple *port/protocol* combinations are separated by a semi-colon. The empty string can be specified to remove all existing MLP shared values. This subcommand argument is optional.

A shared IP address can reduce the total number of IP addresses that are needed on the system, especially when configuring a large number of zones. Unlike the case of the zone-specific IP address, when MLPs are declared on shared IP addresses, only the global zone can receive the incoming network traffic that is destined for the MLP.

- One of the following sets of arguments must be specified for subcommand `add`:

```
-n zonename -l label [-x policymatch=policy-match-level \
-x mlpzone=port/protocol;.... | \
-x mlpshared=port/protocol;.... ]
-h
```

- One of the following sets of arguments must be specified for subcommand `modify`:

```
-n zonename [-l label] [-x policymatch=policy-match-level \
-x mlpzone=port/protocol;.... |\
-x mlpshared=port/protocol;.... ]
-h
```

- One of the following arguments must be specified for subcommand `delete`:

```
-n zonename |
-h
```

- The following argument can be specified for subcommand `list`:

```
-n zonename |
-h
```

使用例

例 1 Adding a New Entry to the Zone Configuration Database

The admin role creates a new zone entry, `public`, with a label of `public`, a policy match level of 1, and a shared MLP port and protocol of 666 and TCP. The administrator is prompted for the admin password.

```
$ /usr/sadm/bin/smtnzonecfg add -- -n public -l public \
-x policymatch=1 -x mlpshared=666/tcp
```

例 2 Modifying an Entry in the Zone Configuration Database

The admin role changes the `public` entry in the `tnzonecfg` database to `needtoknow`. The administrator is prompted for the admin password.

```
$ /usr/sadm/bin/smtnzonecfg modify -- -n public -l needtoknow
```

例 3 Listing the Zone Configuration Database

The admin role lists the entries in the `tnzonecfg` database. The administrator is prompted for the admin password.

```
$ /usr/sadm/bin/smtnzonecfg list --
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` Invalid command syntax. A usage message displays.

`2` An error occurred while executing the command. An error message displays.

ファイル

The following files are used by the `smtnzonecfg` command:

`/etc/security/tsol/tnzonecfg` Trusted zone configuration database.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWmgts
Interface Stability	Committed

関連項目	attributes(5)
注意事項	The functionality described on this manual page is available only if the system is configured with Trusted Extensions.

名前

smtp-notify – ソフトウェアイベントの電子メール通知デーモン

形式

/usr/lib/fm/notify/smtp-notify

機能説明

smtp-notify は、ソフトウェアイベントや障害管理ライフサイクルイベントに登録するデーモンで、SMF サービス構成リポジトリに格納されている一連の通知設定に基づいて電子メール通知を生成します。

電子メール通知設定は、svccfg(1M) を使用して設定されます。

smtp-notify は、サービス FMRI の下でサービス管理機能 smf(5) によって管理されます。

svc:/system/fm/smtp-notify:default

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

プロパティー

次のサービスプロパティーを設定できます。

config/rootdir
: これは astring プロパティーで、デフォルトは / です。設定されている場合は、指定されたルートディレクトリが、smtp-notify によって評価されるすべてのパス名に使用されます。

デフォルトでは、電子メールイベント通知の本文は、Oracle Solaris オペレーティングシステムとともに提供される一連のローカライズされたメッセージテンプレートに基づいて作成されます。smtp-notify は (イベントクラスに基づいて) 適切なテンプレートを検索し、イベントペイロードから要素を使用してそのテンプレートに入力します。

msg_template 通知設定を指定すると、ユーザーが指定したメッセージ本文テンプレートに置き換えることができます。サービスまたはイベントクラスごとの通知設定の詳細については、svccfg(1M) を参照してください。

メッセージ本文テンプレートには、次に示す展開マクロを含めることができます。これらのマクロは、メッセージの送信前に smtp-notify によって展開されます。

```
Macro          Description
-----          -----------
%%              expands to a literal % character
%<HOSTNAME>     expands to the hostname on which the event
                occurred
%<URL>          expands to the URL of the knowledge article
                associated with this event
%<CLASS>        expands to the event class
%<UUID>         expands to the UUID of the event
```

```
%<CODE>         expands to the knowledge article message ID
%<SEVERITY>     expands to the severity of the event
```

SMF サービス状態遷移イベントの場合は、メッセージテンプレート内に次の追加のマクロを指定できます。

```
%<FMRI>          expands to FMRI of the affected service
%<FROM-STATE>    expands to the previous state of the service
%<TO-STATE>      expands to the new state of the service
```

電子メールのフィルタ処理を容易にするために、smtp-notify は、すべてのイベントクラスに対するイベントペイロードの次のコンポーネントのための X ヘッダーを作成します。

```
Header name             Value
-----------             -----
X-FMEV-HOSTNAME         the name of the host on which the
                        event occurred
X-FMEV-CLASS            the event class
X-FMEV-CODE             the Knowledge article message ID
X-FMEV-SEVERITY         the severity of the event
X-FMEV-UUID             the UUID of the event
```

SMF サービス状態遷移イベントにはまた、次の追加の X ヘッダーも含まれます。

```
Header name             Value
-----------             -----
X-FMEV-FMRI             the FMRI of the affected SMF service
X-FMEV-FROM-STATE       the previous state of the service
X-FMEV-TO-STATE         the new state of the service
```

使用例

例 1　通知設定の構成

次のコマンドは、SMF サービス状態遷移イベントの通知設定を構成します。

```
# svccfg -s svc:/system/svc/global:default setnotify -g \
from-online,to-maintenance mailto:admin@somehost.com
```

例 2　電子メール通知の有効化

次のコマンドは、障害管理問題診断イベントの電子メール通知を有効にします。

```
# svccfg setnotify problem-diagnosed mailto:admin@somehost.com
```

例 3　通知設定の一覧表示

次のコマンドは、障害管理問題診断イベントのすべての通知設定を一覧表示します。

```
# svccfg listnotify problem-diagnosed
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management/smtp-notify
インタフェースの安定性	変更の可能性あり

関連項目

svcs(1), svcadm(1M), svccfg(1M), syslogd(1M), attributes(5), smf(5)

名前 sndradm – Sun StorageTek Availability Suite 遠隔ミラーの操作の制御

形式

```
sndradm -I a master shadow bitmap
sndradm -I d master shadow bitmap
sndradm -h usage message
sndradm -v version information
sndradm [options] -e [sndr_set]
sndradm [options] -E [sndr_set]
sndradm [options] -d [sndr_set]
sndradm [options] -D block [sndr_set]
sndradm [options] -D noblock [sndr_set]
sndradm [options] -l [sndr_set]
sndradm [options] -m [sndr_set]
sndradm [options] -m -r [sndr_set]
sndradm [options] -u [sndr_set]
sndradm [options] -u -r [sndr_set]
sndradm [options] -w [sndr_set]
sndradm [options] -H [sndr_set]
sndradm [options] -p [sndr_set]
sndradm [options] -P [sndr_set]
sndradm [options] -q a volume [sndr_set]
sndradm [options] -q d [sndr_set]
sndradm [options] -q r volume [sndr_set]
sndradm [options] -i [sndr_set]
sndradm [options] -a value [sndr_set]
sndradm [options] -A value [sndr_set]
sndradm [options] -F value [sndr_set]
sndradm [options] -W value [sndr_set]
sndradm [options] -R
sndradm [options] -R b p bitmap [sndr_set]
sndradm [options] -R b s bitmap [sndr_set]
sndradm [options] -R C tag [sndr_set]
```

```
sndradm [options] -R g io_groupname [sndr_set]

sndradm [options] -R m sync [sndr_set]

sndradm [options] -R m async [sndr_set]

sndradm [options] -R -f volset-file

sndradm [options] -R r [sndr_set] *
```

機能説明

/usr/sbin/sndradm コマンドは、Sun StorageTek Availability Suite 遠隔ミラーソフトウェアのための管理用コマンド行インタフェースです。遠隔ミラーを使用すると、物理的に別個の異なる Sun サーバー間でディスクをリアルタイムに複製できます。遠隔ミラーは、概念的には RAID 1 のローカルディスクのミラー化方式に似ていますが、より離れた距離で複製操作を実行します。

コマンド行で遠隔ミラーセット (*sndr_set*) を指定しない場合、sndradm は、構成されているすべての遠隔ミラーセットに対して動作します。

sndradm コマンドは、出力 (-p、-P、-i)、ヘルプ (-h)、およびバージョン (-v) を除くすべての操作に対して、Availability Suite ログファイル /var/adm/ds.log (ds.log(4) を参照) 内にエントリを生成します。

オプション

sndradm ユーティリティーは、次のオプションをサポートします。

-f *volset-file*
: 1 つ以上の遠隔ミラーセットの *sndr_set* 情報を含むファイルを、後述する完全に指定されたコマンド行の *sndr_set* と同じ形式で指定します。

-g *io_groupname*
: *io_groupname* に属する遠隔ミラーセットのみに操作を制限します。

  特定のセットの *io_groupname* は、一次ホストと二次ホストの両方で一貫性を持っている必要があります。

-C *tag*
: クラスタ化されたノードでは、*tag* で指定されたクラスタリソースグループまたはディスクグループ名に属する遠隔ミラーセットのみに操作を制限します。このオプションは、クラスタ化されていないシステム上では無効です。

-n
: sndradm を使用して遠隔ミラー操作を開始したあとに、ユーザーに応答を求めません。出力、ヘルプ、およびバージョンを除くすべてのオプションでは、デフォルトの動作で応答を求めます。たとえば、一次ボリュームから二次ボリュームへの完全な同期を開始したあと、遠隔ミラーから「Overwrite secondary with primary? (Y/N) [N]」というメッセージが表示されます。

sndr_set
: 遠隔ミラーセットを指定します。すでに有効になっているセットの場合は、*shost*:*sdev* の形式で set_name を設定できます。次に示すように、完全に指定された遠隔ミラーセットを構成ファイルと同じ形式で指定できます。

phost pdev pbitmap shost sdev sbitmap `ip {sync | async} \`
`[g` *io_groupname*`] [C` *tag*`]`

これらのパラメータについて、次に説明します。

phost
一次ボリュームが存在するサーバーを指定します。

pdev
複製される一次ボリュームパーティションを指定します。フルパス名のみ (たとえば、`/dev/rdsk/c0t1d0s2`) を指定します。

pbitmap
一次パーティションのビットマップ (スコアボードログ) が格納されているボリュームパーティションを指定します。フルパス名のみ (たとえば、`/dev/rdsk/c0t1d0s3`) を指定します。

shost
二次ボリュームが存在するサーバーを指定します。

sdev
二次ボリュームパーティションを指定します。フルパス名のみ (たとえば、`/dev/rdsk/c0t1d0s4`) を指定します。

sbitmap
二次パーティションのビットマップ (スコアボードログ) が格納されているボリュームパーティションを指定します。フルパス名のみ (たとえば、`/dev/rdsk/c0t1d0s5`) を指定します。

`ip`
ネットワーク転送プロトコルを指定します。

`sync | async`
遠隔ミラーの動作モードを指定します。`sync` は、リモートボリュームが更新されるまで I/O 操作が完了したと見なされない遠隔ミラーモードです。`async` は、リモートボリュームが更新される前に一次ホストの I/O 操作が完了したと見なされる遠隔ミラーモードです。

io_groupname
遠隔ミラーセットが属する遠隔ミラー整合グループの名前を指定します。非同期モードでは、書き込み順序が、遠隔ミラー整合グループ内のすべての複製ボリュームに保持される必要があります。これにより、このグループに属する二次ボリューム内に、対応する一次ボリュームの有効なポイントインタイムコピーが含まれるようになります。

既存のセットを整合グループに追加する場合や、あるセットをグループに追加できるようにする場合は、一次ホストと二次ホストの両方で同じグループ名を使用してこのセットを構成する必要があります。

tag
クラスタ内の操作に対して、ローカルデータやビットマップボリュームのディスクグループ名またはリソースタグがボリューム名によって暗黙的に示されていない場合に、そのグループ名またはタグを指定します (たとえば、/dev/rdsk/md/dg/vol と /dev/vx/rdsk/dg/vol は、どちらもディスクグループ名 dg を示します)。遠隔ミラーに指定されたクラスタタグが適切なクラスタリソースグループタグに一致するようにこと、およびクラスタリソースグループが再構成された場合にすべての Availability Suite サービスを最新の状態に維持することは、ユーザーの責任です。クラスタ化されていないシステム上ではクラスタリソースタグを指定できません。

パラメータ

有効な sndradm コマンドでは、次に示すパラメータのいずれかを指定する必要があります。

-I a *master shadow bitmap*
指定されたマスター、シャドウ、およびビットマップを持つ ndr_ii エントリを Availability Suite 構成ファイルに追加します。sndrsyncd(1M) を参照してください。対応するポイントインタイムコピーセットが存在しない場合は、関連するボリュームに対して次の sync コマンドが発行されたときに、そのポイントインタイムコピーセットが有効になります。このポイントインタイムコピーセットが必要なくなった場合は、iiadm -d で無効にできます。iiadm(1M) を参照してください。

-I d *master shadow bitmap*
指定されたマスター、シャドウ、およびビットマップを持つ ndr_ii エントリを Availability Suite 構成ファイルから削除します。ndr_ii 構成エントリを一覧表示するには、dscfg コマンドを使用します。

-a *value*
このセットの自動同期変数の値 (on または off) を指定します。あるセットに対して autosync が要求されると、この機能は、同期操作が要求された時点からこのセットが手動でロギングモードに設定されるまでアクティブになります。このセットが手動でロギングモードに設定されると、autosync 機能はアクティブでなくなり、次に sync が要求されるまで非アクティブのままになります。autosync がアクティブかどうかを確認するには、sndradm -P を使用します。あるセットに対して自動同期が要求されたかどうかを確認するには、dscfg -l の出力内でこのセットに対する「auto=on;」タグを探します。sndrsyncd(1M) を参照してください。

-A *value*
あるセットが非同期モードで複製されているときに非同期キューを処理するために作成できるスレッドの最大数を指定します。デフォルト値は 2 です。

-W *value*
非同期モードで複製されているセットについて、キューに入れることができる書き込みの最大数を指定します。デフォルト値は 4096 です。たとえば、二次ボリュームの書き込み操作が一次ボリュームから 1 回分しか遅れないようにするには、この値を 1 に設定します。

書き込みの最大数の調整は、メモリーベースの非同期 I/O キューを使用しているセットに対してのみ有効です。この値は、ディスクベースの I/O キューが使用されている場合には無視されます。

-D {block | noblock}
ディスクベースのキューの block|noblock 属性を切り替えます。デフォルト設定は block です。I/O のフィル速度がドレイン速度よりも大きく、キューがいっぱいになるまでの時間が十分にある場合、受信 I/O はキュー上にそのための十分な領域が確保されるまでブロックされます。これは、1 つのボリュームであるか同じ整合グループ内の複数のボリュームであるかどうかにかかわらず、書き込み順序を保持するために行われます。noblock が設定されている場合は、受信 I/O でキューがいっぱいになっても I/O はブロックされません。代わりに、このセットがロギングモードに設定され、ディスクキューの内容は無視されます。あとに続く更新同期によって、最新のデータが二次サイトに同期されます。

-F *value*
非同期モードで複製されているセットについて、カーネルメモリー内のキューに入れることができる 512 バイト FBA の最大数を指定します。デフォルト値は 16384 です。

FBA の最大数の調整は、メモリーベースの非同期 I/O キューを使用しているセットに対してのみ有効です。この値は、ディスクベースの I/O キューが使用されている場合は無視されます。

-h
sndradm の使用法の要約を出力します。

-v
遠隔ミラーのバージョン番号を出力します。

-e
このセットに対する遠隔ミラーを有効にして、スコアボードロギングを有効にします。完全な同期が必要なことを示すために、スコアボードが設定されます。このセットの詳細が現在の構成内に保存されます。dscfg(1M) を参照してください。ストレージボリュームドライバに対して、ローカルボリュームとビットマップボリュームが有効になります (sv(7D) を参照)。

-E
このセットに対する遠隔ミラーを有効にして、スコアボードロギングを有効にします。一次ボリュームと二次ボリュームの完全な同期がすでに保証されていることを示すために、スコアボードがクリアーされます。このセットの詳細が現在の

構成内に保存されます。dscfg(1M) を参照してください。ストレージボリュームドライバに対して、ローカルボリュームとビットマップボリュームが有効になります (sv(7D) を参照)。

-d
このセットに対する遠隔ミラーを無効にして、現在の同期操作をすべて停止します。sndradm -d では、一次ボリュームと二次ボリューム間の一時的な違いを追跡するアクティブなスコアボードもすべて破棄されます。

-l
一次ボリュームと二次ボリューム間の遠隔ミラー複製操作とコピー操作を停止し、これらのボリュームに対して独立した遠隔ミラースコアボードロギングを開始します。整合グループ内のすべてのセットが複製中である場合は、二次ボリューム内に、対応する一次ボリュームの有効なポイントインタイムコピーが含まれていることを示します。この条件の下では、ある遠隔ミラーセットがロギングモードに入るとすぐに、rdc カーネルモジュールによって、そのグループ内のほかのすべてのセットが自動的にロギングモードに入ります。これにより、二次ボリューム内に、有効なポイントインタイムコピーが引き続き含まれるようになります。-l パラメータを使用したあとに遠隔ミラーを再開するには、-m パラメータを使用して完全な再同期を実行するか、または -u パラメータを使用して (スコアボードに基づいた) 更新の再同期を実行します。

このオプションは、現在同期が行われている任意のボリュームの二次ボリュームに対しては機能しません。

-w
同期コピーが完了するか、または異常終了するまで待機します。二次システム上で呼び出された場合はただちに復帰します。

-H
指定されたボリュームセットによって使用されるネットワークリンクの健全性について報告します。リンクの健全性は、アクティブまたは非アクティブと報告されます。アクティブは、ネットワークリンクがデータの複製または再同期にアクティブに使用されており、したがって健全な状態にあることを意味します。非アクティブは、ネットワークリンクがデータの複製または再同期にアクティブに使用されておらず、リンクに関する問題を示している可能性があることを意味します。

-p
構成されている遠隔ミラーボリュームまたはセットの一覧を表示します。

-P
構成されている遠隔ミラーボリュームまたはセットの一覧を、詳細情報とともに表示します。(後述の状態の説明を参照してください。)

-q a *volume*
セットまたはグループにディスクキューを追加します。この操作は、そのセットまたはグループがロギングモードの場合に有効です。

`-q d`
セットまたはグループからディスクキューを削除します。この操作は、そのセットまたはグループがロギングモードの場合に有効です。

`-q r` *volume*
グループまたはセットのディスクキューを置き換えます。キューを無効にする操作と同様にセットまたはグループからキューが削除され、キューを追加する操作と同様に新しいディスクキューが追加されます。この操作は、そのセットまたはグループがロギングモードの場合に有効です。

`-i`
構成されている遠隔ミラーボリュームまたはセットの一覧を、*volset-file* と同じ形式で表示します。

`-R`
失敗したビットマップなど遠隔ミラーセットのエラー状態のリセットを試みます。

`-R b p` *bitmap*
遠隔ミラーセットの一次ビットマップを再構成します。このコマンドは、主サーバーと副サーバーの両方で入力するようにしてください。一次ビットマップの再構成は、一度に 1 つのセットに対してのみ実行できます。

`-R b s` *bitmap*
遠隔ミラーセットの二次ビットマップを再構成します。このコマンドは、主サーバーと副サーバーの両方で入力するようにしてください。二次ビットマップの再構成は、一度に 1 つのセットに対してのみ実行できます。

`-R C` *tag*
遠隔ミラーセットのローカルボリュームのクラスタタグまたはディスクグループ名がパス名で示されていない場合に、そのタグまたはグループ名を再構成します。この動作がリモートボリュームに影響を及ぼすことはありません。このパラメータは、クラスタ化されていないシステム上では使用できません。

`-R m {sync | async}`
遠隔ミラーセットの複製モードを再構成します。整合グループに属するセットは、すべて同期またはすべて非同期のどちらかである必要があります。1 つのグループ内でモードを混在させることはできません。

`-R g` *group*
遠隔ミラーセットの整合グループを再構成します。このコマンドは、主サーバーと副サーバーの両方で同じグループ名を使用して入力するようにしてください。

整合グループからセットを削除するには、整合グループを再構成するときに NULL 文字列 (" ") を指定します。

次のパラメータは、主サーバーからのみ発行できます。

-m
一次ボリュームから二次ボリュームへのフルボリュームコピーを開始すると同時に、一次ボリュームから二次ボリュームへの新しい更新の遠隔ミラー複製を有効にします。このパラメータは、一次ボリュームと二次ボリュームが異なる可能性があり、これらのボリュームを増分再同期するためのロギング情報が存在しない場合に使用します。「終了ステータス」を参照してください。

-r
同期を逆方向にして、一次ボリュームが二次ボリュームから同期されるようにします。このパラメータは、-m または -u パラメータとともに使用します。-m -r が設定されると、二次 (ソース) ボリュームから一次 (ターゲット) ボリュームへのフルボリュームコピーを開始しますが、同時に一次 (ソース) ボリュームから二次 (ターゲット) ボリュームへの新しい更新の遠隔ミラー複製を有効にして、ボリュームセットの同期状態が保持されるようにします。-m -r は、一次ボリュームと二次ボリュームの内容が異なる可能性があり、二次ボリュームに目的の内容があるものの、これらのボリュームを (-u を使用して) 増分再同期するためのロギング情報が存在しない場合に使用します。-u -r が指定されると、複製の中断中に保持されていた遠隔ミラーのスコアボードログを使用して、二次 (ソース) ボリュームから一次 (ターゲット) ボリュームを再同期します。次に、一次ボリュームから二次ボリュームへの新しい更新の遠隔ミラー複製を再開して、ボリュームセットの同期状態が保持されるようにします。復元または更新操作中のボリュームセットに対する作業は中止します。これにより、新しい更新の複製が再開される前に一次ボリュームと二次ボリュームが一致するようになります。

-u
遠隔ミラーボリュームセットを更新します。このパラメータにより、遠隔ミラーボリュームセットが再同期されます。遠隔ミラーのスコアボードに変更済みとして記録されているブロックのみが更新されます。一次ボリュームの遠隔ミラー複製を有効にすると同時に、遠隔ミラーのスコアボードログを使用して再同期プロセスを開始し、対応する二次ボリュームが一次ボリュームに一致するようにします。

sndradm -P から返される状態

sndradm -P から返される可能性のある状態を次に示します。

volume failed
ローカルデータボリュームに対する I/O 操作が失敗しました。

bitmap failed
ローカルビットマップボリュームに対する I/O 操作が失敗しました。

disk queue failed
ディスクキューボリュームに対する I/O 操作が失敗しました。

need sync
このボリュームへの同期が中断されました。この同期を完了する (またはポイントインタイムコピーを使用して復元する) 必要があります。どちらかが完了するまで、データフローの方向を変更してはいけません。

need reverse sync
このボリュームへの逆同期が中断されました。この同期を完了する (またはポイントインタイムコピーを使用して復元する) 必要があります。どちらかが完了するまで、データフローの方向を変更してはいけません。

logging
受信した書き込みはビットマップにのみ記録されます。データはリモートサイトに複製されません。need sync、need reverse sync、および queuing はすべてロギングのサブ状態で、書き込みはビットマップに記録されるものの、複製されません。キューイングモード (後述) では、書き込みをビットマップに記録し、非同期フラッシュによる以降の複製のために要求をキューに入れます。

reverse syncing
二次から一次へのコピーが進行中です。

syncing
一次から二次へのコピーが進行中です。

queuing
ディスクキューを使用した通常の非同期複製中、I/O は、非同期フラッシュスレッドによる複製を待機するディスクキューに入れられます。一時的なリンク障害が発生した場合、セットはキューイングモードに移行します。このキューはメモリーベースのキューで使用されるため、破棄されません。代わりに、データがビットマップに記録され、キューに入れられます。リンクが復旧し、sndradm -u が発行されると、(このセットの自動同期を有効にすることによって自動化され、) フラッシュが再開されます。これにより、一時的なリンク停止が発生しても書き込み順序が保持されます。書き込み順序は必要なく、最新のデータのみが必要な場合は、セットを手動でロギングモードに設定して (sndradm -l)、更新同期を発行できます (sndradm -u)。これにより、キュー上のデータが破棄され、ビットマップを使用して高速再同期が実行されます。リンクが復旧する前にキューがいっぱいになった状態で更新同期が発行された場合、キューは破棄され、アプリケーションのハングアップを回避するためにセットがロギングモードに設定されます。

使用例

例 1 遠隔ミラーセットの有効化

次のコマンドは、ホスト example 上で遠隔ミラーの非同期セットを有効にします。ここで、example は一次ホスト、example-remote は二次ホストです。

```
example% sndradm -e example /dev/rdsk/c1t0d0s1 /dev/rdsk/c1t1d0s3 \
example-remote /dev/rdsk/c2t3d0s5 /dev/rdsk/c2t4d0s5 ip async
```

例 2 非同期セットへのディスクキューの追加

次のコマンドは、ディスクキューボリュームを非同期セットに追加します。

```
example% sndradm -q a /dev/rdsk/c1t2d0s3 \
example-remote:/dev/rdsk/c2t3d0s5
```

例 3 非同期セットからのディスクキューの削除

次のコマンドは、ディスクキューボリュームが接続されているセットからディスクキューボリュームを削除します。

```
example% sndradm -q d example-remote:/dev/rdsk/c2t3d0s5
```

例 4 遠隔ミラーセットの無効化

次のコマンドは、ホスト example 上で有効になっている遠隔ミラーセットを無効にします。

```
example% sndradm -d example-remote:/dev/rdsk/c2t3d0s5
```

終了ステータス

0 コマンドが正常に完了しました。

>0 エラーが発生しました。

-m または -u オプションがスクリプト内で実行された場合、これらのいずれかのオプションに続く終了ステータスは、遠隔ミラーセットの現在のステータスには関係なく常に正常終了を返します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-remote-mirror
インタフェースの安定性	確実

関連項目

dscfg(1M), sndrd(1M), sndrsyncd(1M), ds.log(4), rdc.cf(4), attributes(5), sv(7D)

名前

sndrd – 遠隔ミラーデーモン

形式

`/usr/lib/sndrd [-c max_connections] [-l listen_backlog]`

機能説明

sndrd デーモンは、クライアントの遠隔ミラー要求を処理します。このデーモンを実行できるのは、root ユーザーまたは同等の特権を持つユーザーだけです。このデーモンは、実行レベル 2 で自動的に起動されます。sndrd は TCP トランスポートレイヤーを再起動します。

sndrd の起動パラメータを変更する管理者は、root または同等のユーザーとして、/lib/svc/method/svc-rdcsyncd ファイルを編集するのではなく /etc/default/sndr ファイルを変更するようにしてください。sndr(4) を参照してください。

オプション

sndrd デーモンは、次のオプションをサポートします。

-c *max_connections*
: 接続型トランスポートを介してサーバーに許可される接続の最大数を設定します。デフォルトでは、この接続数は 16 です。

-l *listen_backlog*
: 接続型トランスポートを介した RDC TCP の接続キューの長さを設定します。デフォルト値は 10 エントリです。

終了ステータス

0 デーモンが正常に起動されました。

>0 デーモンの起動に失敗しました。

エラー情報は、レベル LOG_ERR で syslog に報告されます。

ファイル

/lib/svc/method/svc-rdcsyncd
: sndrd を起動するためのシェルスクリプト。

/lib/svc/method/svc-rdc
: sndrd を停止するためのシェルスクリプト。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-remote-mirror
インタフェースの安定性	確実

関連項目

svcadm(1M), syslogd(1M), ds.log(4), attributes(5)

注意事項

sndrd デーモンは手動で停止しないでください。sndrd を手動で停止する必要がある場合は、次の手順を実行します。これにより、sndrd デーモンと sndrsyncd デーモンの両方が停止されます。

```
# svcadm disable svc:/system/nws_rdc
```

```
# svcadm disable svc:/system/nws_rdcsyncd
```

sndrd デーモンは手動で起動および再起動しないでください。sndrd を手動で起動する必要がある場合は、次の手順を実行します。これにより、sndrd デーモンと sndrsyncd デーモンの両方が起動されます。

```
# svcadm enable svc:/system/nws_rdc
```

```
# svcadm enable svc:/system/nws_rdcsyncd
```

詳細については、svcadm(1M) を参照してください。

名前

sndrsyncd – Availability Suite 遠隔ミラー更新再同期デーモン

形式

/usr/lib/sndrsyncd

機能説明

sndrsyncd デーモンは、ネットワークまたはマシン障害のあとの更新再同期を自動化し、再同期中に更新されているデータボリュームを保護する必要がある場合にポイントインタイムコピーのコピーを起動します。

このデーモンがカーネルによって通知されるのは、遠隔ミラーによって使用されているネットワークリンクがダウンしたあと、自動同期が有効でそのネットワークリンクを使用しているすべての遠隔ミラーセットを再同期するために、そのネットワークリングが -u オプションを設定して sndradm(1M) コマンドを呼び出したときです。遠隔ミラーセットの自動同期を構成する方法の詳細については、sndradm(1M) を参照してください。

このデーモンは、いずれかの遠隔ミラー再同期が開始または完了したときにも通知されます。Availability Suite 構成ファイルで構成されている場合、このデーモンはポイントインタイムコピーのスナップショットを作成します。副サーバー上では、このデーモンは、ファイルシステムが二次ボリュームに現在マウントされているかどうかをチェックし、そのファイルシステムが現在マウントされている場合は同期の開始を許可しないようカーネルに通知します。

終了ステータス

0　　デーモンが正常に起動されました。

>0　　デーモンの起動に失敗しました。

ファイル

/lib/svc/method/svc-rdcsyncd
　sndrsyncd を起動するためのシェルスクリプト。

/lib/svc/method/svc-rdc
　sndrsyncd を停止するためのシェルスクリプト。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	storage/avs/avs-remote-mirror
インタフェースの安定性	確実

関連項目

iiadm(1M), sndradm(1M), sndrd(1M), svcadm(1M), ds.log(4), rdc.cf(4), attributes(5)

注意事項

sndrsyncd デーモンは手動で停止しないでください。sndrsyncd を手動で停止する必要がある場合は、次の手順を実行します。これにより、sndrd デーモンと sndrsyncd デーモンの両方が停止されます。

```
# svcadm disable svc:/system/nws_rdc

# svcadm disable svc:/system/nws_rdcsyncd
```

sndrsyncd デーモンは手動で起動および再起動しないでください。sndrsyncd を手動で起動する必要がある場合は、次の手順を実行します。これにより、sndrd デーモンと sndrsyncd デーモンの両方が起動されます。

```
# svcadm enable svc:/system/nws_rdc

# svcadm enable svc:/system/nws_rdcsyncd
```

詳細については、svcadm(1M) を参照してください。

名前

snmpdx – Sun Solstice Enterprise Master Agent

形式

```
/usr/lib/snmp/snmpdx [-hy] [-a filename] [-c config-dir]
     [-d debug-level] [-i filename] [-m GROUP -m SPLIT]
     [-o filename] [-p port] [-r filename]
```

機能説明

The Master Agent, `snmpdx`, is the main component of Solstice Enterprise Agent (SEA) technology. It runs as a daemon process and listens to User Datagram Protocol (UDP) port 161 for SNMP requests. The Master Agent also opens another port to receive SNMP trap notifications from various subagents. These traps are forwarded to various managers, as determined by the configuration file.

Upon invocation, `snmpdx` reads its various configuration files and takes appropriate actions by activating subagents, determining the subtree Object Identifier (OID) for various subagents, populating its own Management Information Bases (MIBs), and so forth. The Master Agent invokes subagents, registers subagents, sends requests to subagents, receives responses from subagents, and traps notifications from subagents.

The Master Agent is invoked by the service management facility `smf(5)` at boot time if `svc:/application/management/snmpdx` is enabled (see NOTES) and contents of the resource configuration file `/etc/snmp/conf/snmpdx.rsrc` are non-trivial.

注 – The SMA (Systems Management Agent) is the default SNMP agent in the Solaris operating system. `snmpdx` is Obsolete and may not be supported in a future release of Solaris.

オプション

The following options are supported:

`-a`*filename*	Specify the full path of the access control file used by the Master Agent. The default access control file is `/etc/snmp/conf/snmpdx.acl`.
`-c`*config-dir*	Specify the full path of the directory containing the Master Agent configuration files. The default directory is `/etc/snmp/conf`.
`-d`*debug-level*	Debug. Levels from 0 to 4 are supported, giving various levels of debug information. The default is 0 which means no debug information is given.
`-h`	Help. Print the command line usage.
`-i`*filename*	Specify the full path of the enterprise-name OID map. This file contains the PID used by the Master Agent for recovery after a crash. It contains tuples of the UNIX process ID, port number, resource name, and agent name. The default file is `/var/snmp/snmpdx.st`.
`-m GROUP` | `-m SPLIT`	Specify the mode to use for forwarding of SNMP requests.

`GROUP` Multiple variables can be included in each request from the Master Agent to the subagents. This results in, at mose, one send-request per agent.

	SPLIT　Each variable in the incoming request results in one send-request to each subagent.
	The default is GROUP.
-o*filename*	Specify the full path of the file containing the tuple (enterprise-name, OID). For example, (Sun Microsystems, 1.3.1.6.1.4.32). The Master Agent uses this file as a base for look-up in the trap-filtering and forwarding process. The default file is /etc/snmp/conf/enterprises.oid.
-p*port*	Specify the port number. The default port number is 161.
-r*filename*	Specify the full path of the resource file to be used by the Master Agent. This file stores information about the subagents that the Master Agent invokes and manages. The default resource file is /etc/snmp/conf/snmpdx.rsrc.
-y	Set a recovery indicator to invoke the recovery module. The recovery process discovers which subagents in the previous session are still active; those subagents not active are re-spawned by the Master Agent.

ファイル

/etc/snmp/conf/enterprises.oid	Enterprise-name OID map
/etc/snmp/conf/snmpdx.acl	Access control file
/etc/snmp/conf/snmpdx.rsrc	Resource configuration file
/var/snmp/snmpdx.st	Master Agent status file
/var/snmp/mib/snmpdx.mib	Master Agent MIB file

終了ステータス

The following error values are returned:

0	Successful completion.
non-zero	An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/snmp/sea
Interface Stability	Obsolete

関連項目

attributes(5), smf(5)

注意事項

The snmpdx service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/application/management/snmpdx
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

snmp-notify – ソフトウェアイベントの SNMP 通知デーモン

形式

/usr/lib/fm/notify/snmp-notify

機能説明

snmp-notify は、ソフトウェアイベントや FMA 診断および修復イベントに登録するデーモンで、SMF サービス構成リポジトリに格納されている一連の通知設定に基づいて SNMP トラップ通知を生成します。

SNMP 通知設定は、svccfg(1M) を使用して設定されます。

snmp-notify は、サービス FMRI の下でサービス管理機能 smf(5) によって管理されます。

svc:/system/fm/snmp-notify:default

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

プロパティー

次のサービスプロパティーを設定できます。

config/rootdir
: これは astring プロパティーで、デフォルトは / です。設定されている場合は、指定されたルートディレクトリが、snmp-notify によって評価されるすべてのパス名に使用されます。

使用例

例 1　通知設定の構成

次のコマンドは、SMF サービス状態移行イベントの通知設定を構成します。

```
# svccfg setnotify -g from-online,to-maintenance snmp:active
```

例 2　SNMP 通知の有効化

次のコマンドは、障害管理診断イベントの通知を有効にします。

```
# svccfg setnotify problem-diagnosed snmp:active
```

例 3　通知設定の一覧表示

次のコマンドは、障害管理診断イベントのすべての通知設定を一覧表示します。

```
# svccfg listnotify fma-diagnosis
```

SNMP 通知設定の構成の詳細については、svccfg(1M) を参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/fault-management/snmp-notify

属性タイプ	属性値
インタフェースの安定性	変更の可能性あり

関連項目

svcs(1), svcadm(1M), svccfg(1M), syslogd(1M), attributes(5), smf(5)

名前

snmpXwbemd – SNMP Adapter Subagent for WBEM

形式

`/usr/sadm/lib/wbem/snmpXwbemd [-d] [-h] [-p port]`

機能説明

The `snmpXwbemd` daemon is a subagent in the Web-Based Enterprise Management (WBEM) services package.

This daemon maps the Simple Network Management Protocol (SNMP) requests forwarded by the Solstice Enterprise Agents (SEA) Master Agent `snmpdx(1M)` into one or more equivalent WBEM Common Information Model (CIM) properties and instances. Further, it remaps the response from the CIM Object Manager into a SNMP response, which it passes back to `snmpdx(1M)`.

A mapping file contains the corresponding Object Identifier (OID), class name, property name, and Abstract Syntax Notation 1 (ASN.1) type for each object. You can also create your own mapping file.

オプション

The following options are supported:

`-d` Displays all debug information.

`-h` Displays help by printing the correct command line usage.

`-p` Specifies the port number to use.

オペランド

The following operand is supported:

port Specifies the port number you want to use.

使用例

例 1 An Example of a 050SUNWwbcou.map File

This mapping file that Sun Microsystems provides contains definitions of objects, in this format:

```
#
#pragma ident    "@(#)050SUNWwbcou.map    1.0    01/04/03 SMI"
#
# Copyright (c) 2001 by Sun Microsystems, Inc.
# All rights reserved.
#
# *** Description of contents ***
#
# First non-commented non-blank line contains required Version label.
# Remaining non-commented non-blank lines are considered map entries
# used as described below:
#
# Column 1 - SNMP OID - Uniquely describes an SNMP variable
# Column 2 - CIM Class Name - CIM class associated with this variable
# Column 3 - CIM Property Name - CIM property that maps to SNMP OID variable
# Column 4 - ASN.1 type - SNMP datatype that dictates how data is mapped
#            to/from SNMP requests.  Supported types are: SnmpString, SnmpOid,
```

例 1　An Example of a 050SUNWwbcou.map File　　(続き)

```
#            SnmpTimeticks, SnmpCounter, SnmpInt, SnmpGauge, SnmpIpAddress,
#            SnmpOpaque)
# Column 5 and greater are ignored
#
Version 1.0

1.3.6.1.2.1.1.1.0 Solaris_ComputerSystem Description SnmpString
1.3.6.1.2.1.1.3.0 Solaris_OperatingSystem LastBootUpTime SnmpTimeticks
1.3.6.1.2.1.1.4.0 Solaris_ComputerSystem PrimaryOwnerContact SnmpString
1.3.6.1.2.1.1.5.0 Solaris_ComputerSystem Name SnmpString

1.3.6.1.2.1.25.1.5.0 Solaris_OperatingSystem NumberOfUsers SnmpGauge
1.3.6.1.2.1.25.1.6.0 Solaris_OperatingSystem NumberOfProcesses SnmpGauge
1.3.6.1.2.1.25.1.7.0 Solaris_OperatingSystem MaxNumberOfProcesses SnmpGauge
1.3.6.1.2.1.25.1.2.0 Solaris_OperatingSystem LocalDateTime SnmpString
```

Each definition of an object in this file contains an OID, its corresponding CIM class name, its corresponding CIM property name, and its corresponding ASN.1 type. Each of these elements is separated by a space character.

ファイル

`/var/sadm/wbem/snmp/map/050SUNWwbcou.map`　The SNMP Adapter Subagent for WBEM MIB–2 mapping file that Sun Microsystems provides contains SNMP Management Information Base (MIB) definitions for the CIM instrumentation that SNMP manages.

終了ステータス

The following exit values are returned:

`0`　Successful completion.

`1`　An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes.

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco
CSI	Enabled
Interface Stability	Committed
MT-Level	Safe

関連項目 snmpdx(1M), attributes(5)

名前

snoop – ネットワークパケットの取得および検査

形式

```
snoop [-aqrCDINPSvV] [-t [r | a | d]] [-c maxcount]
 [-d device] [-i filename] [-n filename] [-o filename]
 [-p first [, last]] [-s snaplen] [-x offset [, length]]
 [expression]
```

機能説明

snoop は、データリンクまたは IP インタフェースからパケットを取得し、その内容を表示します。データリンクまたは IP インタフェースが指定されていない場合、snoop は、使用するデータリンクを選択することで、IP トラフィックのために plumb されたデータリンクに優先順位を割り当てます。snoop は、ネットワークからパケットを効率的に取得するために、pfmod(7M) と bufmod(7M) の STREAMS モジュールを使用します。取得したパケットは、受信した順に表示することも、あとで検査するためのファイル (RFC 1761 準拠) に保存することもできます。

snoop は、パケットを単一行の要約形式または詳細な複数行形式で表示できます。要約形式では、特定の VLAN パケットを除いて、もっとも高いレベルのプロトコルに関連したデータのみが表示されます。パケットに VLAN ヘッダーがあり、その VLAN ID がゼロ以外の場合、snoop では、そのパケットは VLAN タグ付きとして表示されます。たとえば、NFS パケットでは NFS 情報だけが表示されます。直前に説明した条件下の VLAN 情報を除いて、基になる RPC、UDP、IP、および Ethernet のフレーム情報は抑制されますが、いずれかの詳細オプションを選択した場合は表示できます。

LDAP や NIS などのネームサービスがない場合、snoop はホスト名を数値の IP アドレスとして表示します。

snoop には、対話型のインタフェースが必要です。

オプション

-C
: カーネルのパケットフィルタまたは snoop 独自のフィルタのいずれかに対するフィルタ式から生成されたコードを一覧表示します。

-D
: 要約行の取得中にドロップしたパケットの数を表示します。

-N
: 取得ファイルから IP アドレスと名前の対照ファイルを作成します。これは、取得ファイルに名前を付ける -i オプションとともに設定する必要があります。アドレスと名前の対照ファイル名は、取得ファイル名に .names を付加したものと同じです。このファイルは、取得するサイトで IP アドレスからホスト名へのマッピングを記録し、取得ファイルの可搬性を高めます。取得ファイルを別の場所で分析する場合は、 .names ファイルを生成します。このフラグを使用すると、パケットは表示されません。

-I *interface*
interface で指定された IP インタフェース (たとえば、lo0) を使用して、ネットワークから IP パケットを取得します。ipadm(1M) コマンドを使用すると、使用可能な IP インタフェースを一覧表示できます。-I オプションと -d オプションを同時に指定することはできません。

-P
プロミスキュアスでないモードでパケットを取得します。ブロードキャスト、マルチキャスト、またはホストマシン宛に送られたパケットだけが表示されます。

-S
要約行のリンクレイヤーフレーム全体のサイズをバイト単位で表示します。

-V
詳細要約モードです。詳細度は、要約モードと詳細モードの中間です。パケット内のもっとも高いレベルのプロトコルの要約行だけを表示するのではなく、パケット内のプロトコルレイヤーごとに要約行を表示します。たとえば、NFS パケットでは、ETHER、IP、UDP、RPC、および NFS レイヤーごとに 1 行ずつ表示します。詳細要約モードの出力は、grep でパイプすることで、目的のパケットを簡単に抽出できます。たとえば、RPC の要約行だけを表示するには、次のように入力します。example# **snoop -i rpc.cap -V | grep RPC**

-a
パケットを /dev/audio で音声で聴きます (警告: うるさくなる可能性があります)。

-c *maxcount*
maxcount パケットを取得したあとで終了します。そうしない場合、残りのディスク容量がなくなるまで、または Control + C で中断するまで取得し続けます。

-d *datalink*
datalink で指定された DLPI データリンク (たとえば、bge0 または net0) を使用して、ネットワークからリンクレイヤーパケットを取得します。dladm(1M)show-link サブコマンドを使用すると、使用可能なデータリンクを一覧表示できます。-d オプションと -I オプションを同時に指定することはできません。

-i *filename*
前に *filename* で取得したパケットを表示します。このオプションを指定しない場合、snoop はネットワークインタフェースからパケットを読み取ります。*filename*.names ファイルが存在する場合、そのファイルは自動的に snoop IP アドレスと名前のマッピングテーブル (-N フラグを参照) にロードされます。

-n *filename*
filename を IP アドレスと名前のマッピングテーブルとして使用します。このファイルは、/etc/hosts ファイル (IP アドレスの次にホスト名) と同じ形式である必要があります。

-o *filename*
取得したパケットを、取得したとおりに *filename* に保存します。(この *filename* を「取得ファイル」と呼びます。)取得ファイルの形式は RFC 1761 に準拠しています。パケットの取得中、ファイルに保存されたパケット数のカウントが表示されます。ファイルに保存せずにパケットのカウントだけを行う場合は、ファイル名を /dev/null とします。

-p *first* [, last]
表示するパケットを取得ファイルから 1 つ以上選択します。ファイルの *first* パケットはパケット番号 1 です。

-q
ネットワークパケットをファイルに取得するときに、パケットカウントを表示しません。これにより、パケット取得のパフォーマンスが向上する場合があります。

-r
IP アドレスを記号名に解決しません。これにより、snoop がパケットを取得および表示している間に、ネットワークトラフィックが生じなくなります。ただし、-n オプションを使用していて、アドレスがマッピングファイルに存在する場合は、それに対応する名前が使用されます。

-s *snaplen*
各パケットの *snaplen* バイトより後ろを切り捨てます。通常は、パケット全体が取得されます。このオプションは、特定のパケットヘッダー情報だけが必要な場合に便利です。パケットの切り捨てはカーネル内で行われるため、ストリームパケットバッファーを効率的に利用できます。これにより、トラフィックが多い期間中のバッファーオーバーフローによってパケットがドロップされる可能性が減少します。また、大量のトレースを取得ファイルに取得するときにディスク容量を節約できます。IP ヘッダーのみ (オプションなし) を取得するには、34 の *snaplen* を使用します。UDP では 42 を、TCP では 54 を使用します。RPC ヘッダーは 80 バイトの *snaplen* で取得できます。NFS ヘッダーは 120 バイトで取得できます。

-t [r | a | d]
タイムスタンプの表示。タイムスタンプの精度は 4 マイクロ秒以内です。デフォルトでは、d (デルタ) 形式 (前のパケットを受信してからの時間) で時間が表示されます。オプション a (絶対的) では、時計時刻になります。オプション r (相対的) では、最初にパケットを表示してからの相対的な時間になります。これを -p オプションとともに使用すると、選択した任意のパケットに対する相対的な時間を表示できます。

-v
詳細モード。パケットヘッダーを数多くの詳細情報とともに出力します。この表示方式では 1 つのパケットについて多数の行を消費するため、選択したパケットに対してのみ使用してください。

-x*offset* [, *length*]
パケットデータを 16 進数および ASCII 形式で表示します。*offset* 値および *length* 値によって、パケットの表示部分を選択します。パケット全体を表示するには、*offset* として 0 を使用します。*length* 値を指定しない場合、パケットの残りの部分が表示されます。

オペランド

expression
パケットをネットワークまたは取得ファイルから選択します。式が真となるパケットのみ選択されます。式が指定されない場合は、真とみなされます。

フィルタ式を指定した場合、snoop は、カーネルパケットフィルタ用または独自の内部フィルタ用のコードを生成します。ネットワークインタフェースでパケットを取得する場合、カーネルパケットフィルタ用のコードが生成されます。このフィルタは、バッファーモジュールのアップストリームであるストリームモジュールとして実装されます。バッファーモジュールは、いっぱいになるまでパケットを蓄積して snoop に渡します。カーネルパケットフィルタは、カーネル内の不要なパケットをパケットバッファーまたは snoop に到達する前に拒否するため、非常に効率的です。カーネルパケットフィルタにはその実装にいくつかの制限があり、フィルタが処理できないフィルタ式を作成することも可能です。この場合、snoop はフィルタの分割を試み、カーネルで可能な限りのフィルタ処理を行います。残りのフィルタ処理は、snoop 用のパケットフィルタによって行われます。-C フラグを使用すると、カーネルのパケットフィルタ用または snoop のパケットフィルタ用に生成されたコードを表示できます。パケットが -i オプションを使用して取得ファイルから読み込まれた場合は、snoop 用のパケットフィルタのみ使用されます。

フィルタの *expression* は、1 つまたは複数のブール型プリミティブで構成されます。これらのプリミティブは、ブール型演算子 (AND、OR、および NOT) と組み合わせることができます。ブール型演算子の通常の優先順位規則が適用されます。これらの演算子の評価の順序は、括弧で制御できます。括弧およびその他のフィルタ式の文字はシェルによって認識されるため、多くの場合フィルタ式を引用符で囲む必要があります。より効率的なフィルタの設定については、例 2 を参照してください。

次にプリミティブを示します。

host *hostname*
発信元アドレスまたは着信先アドレスが hostname のアドレスであれば真。*hostname* 引数は、アドレス表記でもかまいません。名前がほかの式のプリミティブの名前と競合しない場合は、キーワード host を省略できます。たとえば、「pinky」では、ホスト pinky との間で送受信されるパケットが選択され、「pinky and dinky」では、ホスト pinky とホスト dinky との間で交換されるパケットが選択されます。

使用されるアドレスのタイプは、host プリミティブに先行するプリミティブに依存します。使用可能な修飾子は、inet、inet6、ether、または修飾子なし

です。以降では、これら 3 つのプリミティブについて説明します。プリミティブが 1 つもない場合は、「inet host hostname or inet6 host hostname」と同等です。言い換えると、snoop はホスト名と関連付けられた IP アドレスをすべてフィルタ処理しようとします。

inet または *inet6*
後続の `host` プリミティブを変更する修飾子。*inet* の場合、`snoop` は名前の検索によって返された IPv4 アドレスをすべてフィルタ処理しようとします。*inet6* の場合、`snoop` は名前の検索によって返された IPv6 アドレスをすべてフィルタ処理しようとします。

ipaddr、*atalkaddr*、または *etheraddr*
アドレス表記、ドット区切り IP、ドット区切り AppleTalk、およびコロン区切り Ethernet が認識されます。次に例を示します。

- 「`172.16.40.13`」は、この IP を持つすべてのパケットと一致します。
- 「`2::9255:a00:20ff:fe73:6e35`」は、発信元または着信先としてこの IPv6 アドレスを持つすべてのパケットと一致します。
- 「`65281.13`」は、この AppleTalk アドレスを持つすべてのパケットと一致します。
- 「`8:0:20:f:b1:51`」は、発信元または着信先としてこの Ethernet アドレスを持つすべてのパケットと一致します。

文字で始まる Ethernet アドレスは、ホスト名として解釈されます。これを回避するため、アドレスを指定する場合は最初にゼロを付けます。たとえば、Ethernet アドレスが「`aa:0:45:23:52:44`」の場合は、最初に 0 を追加して「`0aa:0:45:23:52:44`」と指定します。

`from` または `src`
後続の `host`、`net`、*ipaddr*、*atalkaddr*、*etheraddr*、`port`、または `rpc` プリミティブを変更して、発信元アドレス、ポート、または RPC 応答とのみ一致する修飾子。

`to` または `dst`
後続の `host`、`net`、*ipaddr*、*atalkaddr*、*etheraddr*、`port`、または `rpc` プリミティブを変更して、着信先アドレス、ポート、または RPC 呼び出しとのみ一致する修飾子。

`ether`
後続の `host` プリミティブを変更して、名前を Ethernet アドレスに解決する修飾子。通常、IP アドレスのマッチングが行われます。このオプションは、IPoIB (IP over InfiniBand) などのメディアではサポートされていません。

`ethertype` *number*
Ethernet タイプフィールドの値が *number* であれば真。*number* が 0x8100 (VLAN) ではなく、パケットが VLAN タグ付きの場合、式はカプセル化した Ethernet タイプと一致します。

`ip`、`ip6`、`arp`、`rarp`、`pppoed`、`pppoes`
パケットが適切な ethertype であれば真。

`vlan`
パケットに `ethertype` VLAN があり、VLAN ID がゼロでなければ真。

`vlan-id` *id*
ID が *id* の ethertype VLAN のパケットであれば真。

`pppoe`
パケットの ethertype が `pppoed` または `pppoes` であれば真。

`broadcast`
パケットがブロードキャストパケットであれば真。Ethernet の `ether[2:4] = 0xffffffff` と同等です。このオプションは、IPoIB (IP over InfiniBand) などのメディアではサポートされていません。

`multicast`
パケットがマルチキャストパケットであれば真。Ethernet の「`ether[0] & 1 = 1`」と同等です。このオプションは、IPoIB (IP over InfiniBand) などのメディアではサポートされていません。

`bootp`、`dhcp`
パケットが、`BOOTPS (67)` の発信元ポートおよび `BOOTPC (68)` の着信先ポート、または `BOOTPC (68)` の発信元ポートおよび `BOOTPS (67)` の着信先ポートのいずれかで断片化されていない IPv4 UDP パケットであれば真。

`dhcp6`
パケットが、`DHCPV6-SERVER (547)` の発信元ポートおよび `DHCPV6-CLIENT (546)` の着信先ポート、または `DHCPV6-CLIENT (546)` の発信元ポートおよび `DHCPV6-SERVER (547)` の着信先ポートのいずれかで断片化されていない IPv6 UDP パケットであれば真。

`apple`
パケットが Apple Ethertalk パケットであれば真。「`ethertype 0x809b or ethertype 0x80f3`」と同等です。

`decnet`
パケットが DECNET パケットであれば真。

`greater` *length*
パケットが *length* より長ければ真。

`less` *length*
パケットが *length* より短ければ真。

`udp`、`tcp`、`icmp`、`icmp6`、`ah`、`esp`
IP または IPv6 プロトコルが適切なタイプであれば真。

net *net*
発信元または着信先の IP アドレスのネットワーク番号が *net* であれば真。from または to 修飾子を使用して、発信元アドレスまたは着信先アドレスでのみネットワーク番号が使われているパケットを選択できます。

port *port*
発信元ポートまたは着信先ポートが *port* であれば真。*port* には、ポート番号または /etc/services から得た名前を使用できます。tcp または udp プリミティブを使用して、TCP または UDP ポートだけを選択できます。from または to 修飾子を使用して、発信元または着信先としてのみ *port* が使われているパケットを選択できます。

rpc *prog* [, *vers* [, proc]]
パケットが *prog* で識別されるプロトコルの RPC 呼び出しパケットまたは応答パケットであれば真。*prog* には、/etc/rpc から得た RPC プロトコルの名前またはプログラム番号を使用できます。*vers* および proc を使用して、プログラムの *version* および *procedure* 番号をさらに修飾できます。たとえば、rpc nfs,2,0 では、NFS NULL 手続きの呼び出しと応答がすべて選択されます。to または from 修飾子を使用して、呼び出しパケットだけまたは応答パケットだけを選択できます。

zone *zoneid*
zoneid が、ipnet デバイス上で受信されたパケットの発信元または着信先 *zoneid* のどちらかに一致すれば、真です。

ldap
パケットがポート 389 の LDAP パケットであれば真。

gateway *host*
パケットで *host* がゲートウェイとして使用された場合、つまり Ethernet の発信元または着信先アドレスが IP アドレスではなく *host* であった場合は真。「ether host *host* and not host *host*」と同等です。

nofrag
パケットが断片化されていないか、一連の IP フラグメントの最初のものであれば真。ip[6:2] & 0x1fff = 0 と同等です。

expr relop expr
relop が >、<、>=、<=、=、!= のいずれかで、expr が数字、パケットフィールドセレクタ、length プリミティブ、および算術演算子 +、−、*、&、|、^、% で構成される算術式である関係が成立すれば真。expr 内の算術演算子は関係演算子より前に評価され、通常の優先順位規則が算術演算子間で適用されます (加算する前に乗算する、など)。括弧を使用して評価の順序を制御できます。パケット内のフィールドの値を使用するには、次の構文を使用します。

base[expr [: size]]

ここで expr は、*base* オフセットからパケットへのオフセットの値を評価します。base オフセットには、ether、ip、ip6、udp、tcp、または icmp を使用で

きます。size 値は、フィールドのサイズを指定します。指定されなかった場合は、1 が使用されます。その他の有効な値は 2 と 4 です。次に例を示します。

```
ether[0] & 1 = 1
```

は、multicast と同等です。

```
ether[2:4] = 0xffffffff
```

は、broadcast と同等です。

```
ip[ip[0] & 0xf * 4 : 2] = 2049
```

は、udp[0:2] = 2049 と同等です。

```
ip[0] & 0xf > 5
```

は、オプション付きの IP パケットを選択します。

```
ip[6:2] & 0x1fff = 0
```

は、IP フラグメントを排除します。

```
udp and ip[6:2]&0x1fff = 0 and udp[6:2] != 0
```

は、UDP チェックサムのあるすべてのパケットを検索します。

length プリミティブを使用して、パケットの長さを取得できます。たとえば、「length > 60」は「greater 60」と同等であり、「ether[length − 1]」では、パケット内の最後のバイトの値を取得します。

and
: 2 つのブール型値の間で論理 AND 演算を実行します。2 つのブール型式を並べた場合は AND 演算の意味になります。たとえば、「dinky pinky」は「dinky AND pinky」と同じです。

or または ,
: 2 つのブール型値の間で論理 OR 演算を実行します。代わりにコンマを使用することもできます。たとえば、「dinky,pinky」は「dinky OR pinky」と同じです。

not または !
: 後続のブール型値に対して論理 NOT 演算を実行します。この演算子は、AND または OR より前に評価されます。

slp
: パケットが SLP パケットであれば真。

sctp
: パケットが SCTP パケットであれば真。

ospf
: パケットが OSPF パケットであれば真。

使用例

例 1　snoop コマンドの使用

すべてのパケットを取得し、受信した順に表示します。

```
example# snoop
```

ホスト funky が発信元または着信先であるパケットを取得し、受信した順に表示します。

```
example# snoop funky
```

funky と pinky の間のパケットを取得し、ファイルに保存します。その後、最初に取得したパケットからの相対的な時間 (秒単位) を使ってパケットを検査します。

```
example# snoop -o cap funky pinky
example# snoop -i cap -t r | more
```

別の取得ファイル内の選択したパケットを表示するには:

```
example# snoop -i pkts -p 99,108
 99   0.0027   boutique -> sunroof     NFS C GETATTR FH=8E6
100   0.0046   sunroof -> boutique     NFS R GETATTR OK
101   0.0080   boutique -> sunroof NFS C RENAME FH=8E6C MTra00192 to .nfs08
102   0.0102   marmot -> viper        NFS C LOOKUP FH=561E screen.r.13.i386
103   0.0072   viper -> marmot       NFS R LOOKUP No such file or directory
104   0.0085   bugbomb -> sunroof    RLOGIN C PORT=1023 h
105   0.0005   kandinsky -> sparky    RSTAT C Get Statistics
106   0.0004   beeblebrox -> sunroof  NFS C GETATTR FH=0307
107   0.0021   sparky -> kandinsky    RSTAT R
108   0.0073   office -> jeremiah      NFS C READ FH=2584 at 40960 for 8192
```

パケット 101 をさらに詳細に表示するには:

```
example# snoop -i pkts -v -p101
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Packet 101 arrived at 16:09:53.59
ETHER:  Packet size = 210 bytes
ETHER:  Destination = 8:0:20:1:3d:94, Sun
ETHER:  Source      = 8:0:69:1:5f:e,  Silicon Graphics
ETHER:  Ethertype = 0800 (IP)
ETHER:
IP:   ----- IP Header -----
IP:
IP:   Version = 4, header length = 20 bytes
IP:   Type of service = 00
IP:         ..0. .... = routine
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:   Total length = 196 bytes
```

例 1　snoop コマンドの使用　　*(続き)*

```
IP:   Identification 19846
IP:   Flags = 0X
IP:   .0.. .... = may fragment
IP:   ..0. .... = more fragments
IP:   Fragment offset = 0 bytes
IP:   Time to live = 255 seconds/hops
IP:   Protocol = 17 (UDP)
IP:   Header checksum = 18DC
IP:   Source address = 172.16.40.222, boutique
IP:   Destination address = 172.16.40.200, sunroof
IP:
UDP:  ----- UDP Header -----
UDP:
UDP:  Source port = 1023
UDP:  Destination port = 2049 (Sun RPC)
UDP:  Length = 176
UDP:  Checksum = 0
UDP:
RPC:  ----- SUN RPC Header -----
RPC:
RPC:  Transaction id = 665905
RPC:  Type = 0 (Call)
RPC:  RPC version = 2
RPC:  Program = 100003 (NFS), version = 2, procedure = 1
RPC:  Credentials: Flavor = 1 (Unix), len = 32 bytes
RPC:     Time = 06-Mar-90 07:26:58
RPC:     Hostname = boutique
RPC:     Uid = 0, Gid = 1
RPC:     Groups = 1
RPC:  Verifier   : Flavor = 0 (None), len = 0 bytes
RPC:
NFS:  ----- SUN NFS -----
NFS:
NFS:  Proc = 11 (Rename)
NFS:  File handle = 00001643000000010008000030 5A1C47
NFS:                597A000000080000204631 4AFC450000
NFS:  File name = MTra00192
NFS:  File handle = 00001643000000010008000030 5A1C47
NFS:                597A000000080000204631 4AFC450000
NFS:  File name = .nfs08
NFS:
```

sunroof と boutique の間の NFS パケットだけを表示するには、次のように入力します。

```
example# snoop -i pkts rpc nfs and sunroof and boutique
1   0.0000   boutique -> sunroof    NFS C GETATTR FH=8E6C
```

例 1 snoop コマンドの使用 *(続き)*

```
2   0.0046    sunroof -> boutique   NFS R GETATTR OK
3   0.0080   boutique -> sunroof   NFS C RENAME FH=8E6C MTra00192 to .nfs08
```

これらのパケットを新規取得ファイルに保存するには:

```
example# snoop -i pkts -o pkts.nfs rpc nfs sunroof boutique
```

カプセル化されたパケットを表示するために、カプセル化を示す指示子があります。

```
example# snoop ip-in-ip
sunroof -> boutique ICMP Echo request    (1 encap)
```

-V をカプセル化されたパケットに使用する場合は、次のようになります。

```
example# snoop -V ip-in-ip
sunroof -> boutique  ETHER Type=0800 (IP), size = 118 bytes
sunroof -> boutique  IP D=172.16.40.222 S=172.16.40.200 LEN=104, ID=27497
sunroof -> boutique  IP  D=10.1.1.2 S=10.1.1.1 LEN=84, ID=27497
sunroof -> boutique  ICMP Echo request
```

例 2 より効率的なフィルタの設定

より効率的なフィルタを設定するには、次のフィルタを式の末尾の方で使用して、式の最初の部分をカーネルで設定できるようにしてください。greater、less、port、rpc、nofrag、および relop です。カーネルで設定できないこれらのプリミティブを使用すると、OR の存在によってフィルタ処理の分割が難しくなります。代わりに、括弧を使って、OR を適用すべきプリミティブを強制します。

port 80 でタイプが tcp または udp の funky と pinky の間のパケットを取得するには、次のように入力します。

```
example# snoop funky and pinky and port 80 and tcp or udp
```

プリミティブ port はカーネルフィルタによって処理することができず、式には OR も存在するため、より効率的にフィルタ処理を行う方法は、OR を式の末尾に移動し、括弧を使って tcp と udp の間に OR を適用することです。

```
example# snoop funky and pinky and (tcp or udp) and port 80
```

終了ステータス

0 正常終了。

1 エラーが発生しました。

ファイル

/dev/audio	システムの主なオーディオデバイスへのシンボリックリンク。
/dev/null	NULL ファイル。
/etc/hosts	ホスト名データベース。

/etc/rpc　RPC プログラム番号データベース。

/etc/services　インターネットサービスとエイリアス。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/network-clients
インタフェースの安定性	廃止

関連項目

dladm(1M), ipadm(1M), netstat(1M), hosts(4), rpc(4), services(4), attributes(5), audio(7I), ipnet(7D), bufmod(7M), dlpi(7P), pfmod(7M)

『RFC 1761, Snoop Version 2 Packet Capture File Format』、Callaghan, B. および Gilligan, R. 共著、Network Working Group 発行、1995 年 2 月。

警告

リアルタイムにパケットを解釈するための処理オーバーヘッドは非常に大きくなります。その結果、ドロップされるパケットの数も多くなる可能性があります。取得の信頼性を高めるためには、-o オプションを使用して raw パケットをファイルに出力し、オフラインでパケットを分析してください。

フィルタ処理しないでパケットを取得すると、特に取得したパケットをリアルタイムで解釈する場合は、ホストコンピュータに大きな処理負荷がかかります。この処理負荷は、詳細オプションを使用するとさらに増加します。snoop を頻繁に使用するとほかの処理に対してコンピューティングリソースが提供されない可能性があるため、本稼働サーバーでは使用しないでください。snoop を頻繁に使用するのは、専用のコンピュータに限定してください。

snoop は IP フラグメントを再構成しません。上のレベルのプロトコルの解釈は、最初の IP フラグメントの末尾で停止します。

snoop を使用する際の派生的な効果として、追加のパケットが生成される場合があります。たとえば、ネットワークネームサービス (NIS) を使用して、IP アドレスをホスト名に変換して表示する場合があります。あとで表示するためにファイルに取得すると、取得のセッションが完了するまでアドレスと名前のマッピングを延期できます。また、NFS でマウントされたファイルへの取得によっても追加のパケットが生成される場合があります。

snaplen (-s オプション) を小さい値に設定すると、上のレベルのプロトコルの解釈に必要なヘッダー情報が削除される場合があります。切り取られる正確な値は、使用するネットワークとプロトコルによって異なります。10 Mbps の Ethernet で UDP を使用する NFS バージョン 2 トラフィックの場合は、snaplen を 150 バイト未満に設定しないでください。100 Mbps の Ethernet で TCP を使用する NFS バージョン 3 トラフィックの場合は、snaplen を 250 バイト以上にしてください。

snoop は、RPC 応答を完全に解釈するために、RPC 要求からの情報を必要とします。取得ファイルまたはパケット範囲の RPC 応答に、先行する要求がない場合は、RPC 応答のヘッダーだけが表示されます。

名前

soconfig – configure transport providers for use by sockets

形式

`/usr/sbin/soconfig -d` *dir*

`/usr/sbin/soconfig -f` *file*

`/usr/sbin/soconfig` *family type protocol* `[`*module* `|` *path*`]`

`/usr/sbin/soconfig -l [-np]`

機能説明

The `soconfig` utility configures the transport provider driver for use with sockets. It specifies how the family, type, and protocol parameters in the `socket(3SOCKET)` call are mapped to the name of a transport provider such as `/dev/tcp`. This utility can be used to add an additional mapping or remove a previous mapping.

The `init(1M)` utility uses `soconfig` with the `sock2path.d(4)` file during the booting sequence.

オプション

The following options are supported:

`-d` *dir*
: Set up the `soconfig` configuration for each driver according to the information stored in the files in *dir*.

`-f` *file*
: Set up the `soconfig` configuration for each driver according to the information stored in *file*. A `soconfig` file consists of lines of at least the first three fields listed below, separated by spaces:

  family type protocol [*module* | *path*]

  These fields are described in the `OPERANDS` section below.

  An example of *file* can be found in the `EXAMPLES` section below.

`-l [-np]`
: Dump the in-kernel socket configuration table. This option has the following suboptions:

  `-n`
  : Do not convert socket family, type, and protocol to string representation.

  `-p`
  : Display machine-parseable output.

オペランド

The following operands are supported:

family
: The protocol family as listed in the `/usr/include/sys/socket.h` file, expressed as an integer.

type
: The socket type as listed in the `/usr/include/sys/socket.h` file, expressed as an integer.

protocol
The protocol number as specified in the family-specific `include` file, expressed as an integer. For example, for `AF_INET` this number is specified in `/usr/include/netinet/in.h`. An unspecified protocol number is denoted with the value zero.

module | *path*
The module name or path name of a device that corresponds to the transport provider, such as `tcp` or `/dev/tcp`. Modules must reside in `kernel/socketmod`. A device name must begin with `/dev`. If this parameter is specified, the configuration will be added for the specified family, type, and protocol. If this parameter is not specified, the configuration will be removed.

使用例

例 1 Using soconfig

The following example sets up a module for family `AF_INET` and type `SOCK_STREAM`:

```
example# soconfig 2 2 0 tcp
```

The following example sets up `/dev/tcp` for family `AF_INET` and type `SOCK_STREAM`:

```
example# soconfig 2 2 0 /dev/tcp
```

The following is a sample file used with the `-f` option. Comment lines begin with a hash mark (#):

```
#   Family  Type  Protocol   Module | Path
      2       2       0          tcp
      2       2       6          tcp

      2       1       0          udp
      2       1       17         udp

      1       2       0          /dev/ticotsord
      1       1       0          /dev/ticlts

      2       4       0          icmp
```

ファイル

`/etc/sock2path.d`
Directory containing files with mappings from sockets to transport providers.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目 init(1M), sock2path.d(4), attributes(5)

『Network Interface Guide』

名前

soladdapp – add an application to the Solstice application registry

形式

```
/usr/snadm/bin/soladdapp [-r registry] -n name -i icon -e executable
    [args]
```

機能説明

`soladdapp` adds an application to the Solstice application registry. After it is added, the application is displayed in the Solstice Launcher main window (see `solstice(1M)`).

オプション

`-r` *registry*	Define the full path name of the Solstice registry file.
`-n` *name*	Define the name of the tool to be registered.
`-i` *icon*	Define the full path name of the tool icon.
`-e` *executable*	Define the full path name of the tool.
args	Specify any arguments to use with the tool.

When executed without options, `soladdapp` uses `/opt/SUNWadm/etc/.solstice_registry` (the default registry path).

戻り値

`0`	on success
`1`	on failure
`2`	if the registry is locked
`3`	if the entry is a duplicate.

使用例

例 1 A sample display of the soladdapp command.

The following adds an application called Disk Manager to the Solstice application registry for display in the Solstice Launcher main window.

```
# soladdapp -r /opt/SUNWadm/etc/.solstice_registry -n "Disk Manager"
-i /opt/SUNWdsk/etc/diskmgr.xpm -e /opt/SUNWdsk/bin/diskmgr
```

ファイル

`/opt/SUNWadm/etc/.solstice_registry`	The default registry path.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsadml

関連項目

`soldelapp(1M)`, `solstice(1M)`, `attributes(5)`

注意事項

Globally registered applications are used by local and remote users sharing the software in a particular /opt directory. They can be added only using `soladdapp`.

名前

soldelapp – remove an application from the Solstice application registry

形式

`/usr/snadm/bin/soldelapp [-r` *registry*`] -n` *name*

機能説明

`soldelapp` removes an application from the Solstice application registry. After removal, the application is no longer displayed in the Solstice Launcher main window (see `solstice(1M)`).

オプション

`-r` *registry* Define the full path name of the Solstice registry file.

`-n` *name* Define the name of the tool to be removed.

When executed without options, `soldelapp` uses `/opt/SUNWadm/etc/.solstice_registry` (the default registry path).

戻り値

`0` on success

`1` on failure

`2` if the registry is locked

`3` if *name* is not found in the registry

`4` if the named registry or default registry is not found

使用例

例 1 A sample display of the soldelapp command.

The following removes an application called Disk Manager from the Solstice application registry and the Solstice Launcher main window.

```
# soldelapp -r /opt/SUNWadm/etc/.solstice_registry -n "Disk Manager"
```

ファイル

`/opt/SUNWadm/etc/.solstice_registry` The default registry file.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsadml

関連項目

`soladdapp(1M)`, `solstice(1M)`, `attributes(5)`

注意事項

Globally registered applications are used by local and remote users sharing the software in a particular `/opt` directory. They can be removed only using `soldelapp`.

名前

solstice – access system administration tools with a graphical user interface

形式

`/bin/solstice`

機能説明

`solstice` used on a system presents the `Solstice Launcher`, a graphical user interface that provides access to the Solstice AdminSuite product family of system administration tools. The tools that appear in the launcher depend on what Solstice products you installed on your system.

Help is available by using the Help button.

使用法

The `Solstice Launcher` allows you to do the following tasks:

Launch applications	Use the `Solstice Launcher` to launch system administration tools.
Register applications	Use the `Solstice Launcher` to add and register applications locally with the launcher.
Remove applications	Use the `Solstice Launcher` to remove locally registered applications.
Customize application properties	Use the `Solstice Launcher` to show, hide, or remove applications in the launcher, reorder the icons, change the launcher window width, modify applications properties, and add applications.

ファイル

`/$HOME/.solstice_registry` Local registry information.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsadml

関連項目

`soladdapp(1M)`, `soldelapp(1M)`, `attributes(5)`

注意事項

The `Solstice Launcher` adds or removes local applications that are private to the user (not local to the system) only. The properties of globally registered applications that are used by local and remote users sharing the software from a particular `/opt` directory cannot be modified from the `Solstice Launcher`. To register global applications for use by local and remote users, use the `soladdapp(1M)` command. To remove globally registered applications, use the `soldelapp(1M)` command.

名前

sppptun – PPP tunneling driver utility

形式

```
sppptun plumb
sppptun plumb [-s sap] protocol device
sppptun unplumb interface
sppptun query
```

機能説明

The `sppptun` utility is used to configure and query the Solaris PPP tunneling device driver, `/dev/sppptun`. Currently, only PPP over Ethernet (PPPoE) is supported, so the `plumb` and `unplumb` arguments are used to specify Ethernet interfaces that are to be used for PPPoE, and the `query` option lists the plumbed interfaces.

The use of `sppptun` to add interfaces is similar to the use of `ifconfig(1M)` to add interfaces to IP. The plumbing is done once for each interface, preferably at system start-up time, and is not normally manipulated on a running system. If multiple instances of PPP are run over a single interface, they share the plumbing to that interface. Plumbing for each session is not required (and not possible for PPPoE).

The proper way to plumb interfaces for PPPoE is to list the interfaces, one per line, in the `/etc/ppp/pppoe.if` file. If alternate Ethertype (SAP) values are necessary, then include the PPPoE Session and Discovery Stage values as hexadecimal on the same line. The line format is:

interface [*session* [*discovery*]]

The defaults are the Ethertypes specified in RFC 2516, and most users should not need to set these values. See the examples for one possible use.

使用法

`sppptun plumb`	When specified with no additional arguments, the plumb argument lists the protocols that are supported by the utility. These are the strings that are used as the *protocol* argument below.
`sppptun plumb [-s` *sap*`]` *protocol device*	This plumbs a new interface into the driver. The *protocol* parameter is `pppoe` for the PPP-carrying "Session Stage" connection or `pppoed` for the PPPoE "Discovery Stage" connection. Both connections must be present for each Ethernet interface that is to be used for PPPoE. The *device* parameter is the path name of the Ethernet interface to use (use `ifconfig(1M)` to list available devices). If the path begins with `/dev/`, then this portion may be omitted. The `-s` *sap* option can be specified to use an alternate Ethertype (SAP) for the selected protocol. The *sap* value must be given in

hexadecimal. Some access servers use Ethertypes for PPPoE different from those in RFC 2516. The defaults are 8864 for `pppoe` and 8863 for `pppoed`.

`sppptun unplumb` *interface*

This removes an existing interface from the driver and terminates any PPP sessions that were using the interface. The *interface* parameter is the name of the interface as reported when the interface was plumbed.

`sppptun query`

Displays the canonical names of all interfaces plumbed into the `/dev/sppptun` device driver.

使用例

例 1 Setting up to Use PPPoE on hme0

Plumb the `hme0` interface.

```
# sppptun plumb pppoed hme0
hme0:pppoed
# sppptun plumb pppoe hme0
hme0:pppoe
```

Remove the `hme0` interface.

```
# sppptun unplumb hme0:pppoed
# sppptun unplumb hme0:pppoe
```

例 2 Script to Remove All Plumbed Interfaces

```
#!/bin/sh
for intf in ‘sppptun query‘
do
        sppptun unplumb $intf
done
```

例 3 Interoperating with 3COM HomeConnect Dual Link ADSL

```
# dladm show-link
LINK        CLASS    MTU    STATE    OVER
nge0        phys     1500   up       --
# echo nge0 3c13 3c12 > /etc/ppp/pppoe.if
```

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` One or more errors occurred.

ファイル

`/etc/ppp/pppoe.if`	list of Ethernet interfaces to be plumbed at boot time
`/usr/sbin/sppptun`	executable command
`/dev/sppptun`	Solaris PPP tunneling device driver

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/ppp/tunnel

関連項目

`pppd(1M)`, `pppoec(1M)`, `pppoed(1M)`, `sppptun(7M)`

RFC 2516, Method for Transmitting PPP Over Ethernet (PPPoE), Mamakos et al, February 1999

名前

spray – spray packets

形式

```
/usr/sbin/spray [-c count] [-d delay] [-l length]
     [-t nettype] host
```

機能説明

`spray` sends a one-way stream of packets to *host* using RPC, and reports how many were received, as well as the transfer rate. The *host* argument can be either a name or an Internet address.

`spray` is not useful as a networking benchmark, as it uses unreliable connectionless transports, UDP for example. `spray` can report a large number of packets dropped when the drops were caused by `spray` sending packets faster than they can be buffered locally, that is, before the packets get to the network medium.

オプション

`-c` *count*
: Specify how many packets to send. The default value of *count* is the number of packets required to make the total stream size 100000 bytes.

`-d` *delay*
: Specify how many microseconds to pause between sending each packet. The default is `0`.

`-l` *length*
: The *length* parameter is the numbers of bytes in the Ethernet packet that holds the RPC call message. Since the data is encoded using XDR, and XDR only deals with 32 bit quantities, not all values of *length* are possible, and `spray` rounds up to the nearest possible value. When *length* is greater than 1514, then the RPC call can no longer be encapsulated in one Ethernet packet, so the *length* field no longer has a simple correspondence to Ethernet packet size. The default value of *length* is `86 bytes`, the size of the RPC and UDP headers.

`-t` *nettype*
: Specify class of transports. Defaults to `netpath`. See `rpc(3NSL)` for a description of supported classes.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

関連項目

`rpc(3NSL)`, `attributes(5)`

名前

srptadm – SRP ターゲットの管理

形式

```
srptadm modify-target [-e, --enable] [-d, --disable]
     [-r, --reset] hca_guid

srptadm list-target [hca_guid]

srptadm modify-defaults [-e, --enable] [-d, --disable]

srptadm list-defaults
```

機能説明

srptadm コマンドは、stmfadm(1M) と libstmf(3LIB) で説明されている SCSI ターゲットモードフレームワーク (SCSI Target Mode Framework、STMF) 内部の SRP (SCSI RDMA Protocol) ターゲットポートを管理します。これにより、SRP イニシエータが SRP プロトコルを使用して SCSI ターゲットモードフレームワーク (STMF) 論理ユニットにアクセスすることが可能になります。

srptadm は、独自のオプションとオペランドを備えたサブコマンドのセットとして実装されます。これらのサブコマンドについては、以降の節でサブコマンドごとに説明します。

サブコマンド

list-defaults
: デフォルトプロパティーの情報を一覧表示します。このサブコマンドにオプションはありません。

list-target [*hca_guid*]
: *hca_guid* が指定されている場合は、ターゲット HCA のプロパティーを一覧表示します。そうでない場合は、すべての HCA のプロパティーが一覧表示されます。

modify-defaults [-e, --enable] [-d, --disable]
: デフォルトのパラメータを変更します。

  -e, --enable
  : modify-target で明示的に無効にされていないすべての HCA に対して SRP ターゲットの作成を有効にします。

  -d, --disable
  : modify-target で明示的に有効にされていないすべての HCA に対して SRP ターゲットの作成を無効にします。

modify-target [-e, --enable] [-d, --disable] [-r, --reset *hca_guid*
: 指定された HCA の SRP ターゲットプロパティーを設定します。

  -e, --enable
  : この HCA での SRP ターゲットの作成を有効にします。

  -d, --disable
  : この HCA での SRP ターゲットの作成を無効にします。

`-r, --reset`
HCA 固有の情報をクリアーし、デフォルトにリセットします。この HCA に SRP ターゲットが関連付けられている場合、その SRP ターゲットは、SRP ターゲットの SMF サービスが再起動されるまで変更されません。

オペランド

list-target サブコマンドと modify-target サブコマンドには、次のオペランドがあります。

hca_guid
SRP ターゲットサービスを提供できる、このシステム上の InfiniBand ホストチャネルアダプタ (HCA) の GUID。この GUID は、次のいずれかの形式である必要があります。

`3BA000100CD18`	基本の 16 進数形式。
`0003BA000100CD18`	先頭にゼロが付けられた基本の 16 進数形式。
`hca:3BA000100CD18`	`cfgadm(1M)` の形式。
`eui.0003BA000100CD18`	EUI の形式。

使用法

SRP サービスが `online` のときにデフォルトの状態が変更されても、既存のターゲットの状態は、そのサービスが再起動されるまで変更されません。

SRP ターゲットサービスが `online` の場合、ターゲットの状態の変更はただちに有効になります。`disabled` に設定されたターゲットは、オフライン状態になって削除されます。`enabled` に設定されたターゲットは、ただちに作成されます。

使用例

例 1 デフォルトプロパティーの一覧表示

次のコマンドは、デフォルトの SRP ターゲットサービスプロパティーを一覧表示します。

```
# srptadm list-defaults
```

例 2 デフォルトの動作の変更

次のコマンドは、SRP ターゲットサービスのデフォルトの動作を変更し、そのサービスが有効になったときに SRP ターゲットが作成されないようにします。

```
# srptadm modify-defaults -d
```

例 3 特定の HCA のプロパティーの一覧表示

次のコマンドは、特定の HCA の SRP ターゲットプロパティーを一覧表示します。

```
# cfgadm | grep hca
hca:3BA000100CD18              IB-HCA       connected    configured   ok
hca:3BA000100D030              IB-HCA       connected    configured   ok

# srptadm list-target hca:3BA000100CD18
```

例 3 特定の HCA のプロパティーの一覧表示 (続き)

```
Target HCA 3BA000100CD18:
    Enabled             : true
    SRP Target Name     : eui.0003BA000100CD18
    Operational Status  : online
```

例 4 特定の HCA のサービスの無効化

次のコマンドは、特定の HCA の SRP ターゲットサービスを無効にします。

```
# srptadm modify-target -d 3BA000100CD18
```

その変更を確認するには、`list-target` コマンドを使用します。

```
# srptadm list-target hca:3BA000100CD18
Target HCA 3BA000100CD18:
    Enabled             : false
    SRP Target Name     : eui.0003BA000100CD18
    Operational Status  : -
```

例 5 ターゲットサービスの再有効化

次のコマンドシーケンスは SRP ターゲットサービスを再度有効にし、SRP と STMF のターゲットプロパティーを表示します。

```
# srptadm modify-target -e 3BA000100CD18
# srptadm list-target hca:3BA000100CD18
Target HCA 3BA000100CD18:
    Enabled             : true
    SRP Target Name     : eui.0003BA000100CD18
    Operational Status  : online

# stmfadm list-target -v eui.0003BA000100CD18
Target: eui.0003BA000100CD18
    Operational Status: Online
    Provider Name     : srpt
    Alias             : -
    Protocol          : SRP
    Sessions          : 0
```

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/storage/scsi-rdma/scsi-rdma-target
インタフェースの安定性	確実

関連項目 cfgadm(1M), stmfadm(1M), libsrpt(3LIB), libstmf(3LIB), attributes(5), srpt(7D)

名前

sshd – secure shell daemon

形式

```
sshd [-deiqtD46] [-b bits] [-f config_file]
     [-g login_grace_time] [-h host_key_file]
     [-h PKCS#11 URI]
     [-p port] [-V client_protocol_id]
```

機能説明

The sshd (Secure Shell daemon) is the daemon program for ssh(1). Together these programs replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. The programs are intended to be as easy to install and use as possible.

sshd is the daemon that listens for connections from clients. It forks a new daemon for each incoming connection. The forked daemons handle key exchange, encryption, authentication, command execution, and data exchange.

This implementation of sshd supports SSH protocol version 2 only.

SSH Protocol Version 1

This protocol version is no longer supported in the daemon program. It is supported only in the client. See ssh(1) manual page for more information.

SSH Protocol Version 2

Each host has a host-specific DSA/RSA key used to identify the host. Forward security is provided through a Diffie-Hellman key agreement. This key agreement results in a shared session key. The rest of the session is encrypted using a symmetric cipher, currently 128/192/256-bit AES CBC or CTR, 128/256-bit ARCFOUR, Blowfish, or 3DES.. The client selects the encryption algorithm to use from those offered by the server. Additionally, session integrity is provided through a cryptographic message authentication code (hmac-sha1 or hmac-md5).

Protocol version 2 provides a public key based user authentication method (PubKeyAuthentication) GSS–API based user authentication, conventional password authentication, and a generic prompt/reply protocol for password-based authentication.

Command Execution and Data Forwarding

If the client successfully authenticates itself, a dialog for preparing the session is entered. At this time the client can request things like allocating a pseudo-tty, forwarding X11 connections, forwarding TCP/IP connections, or forwarding the authentication agent connection over the secure channel.

Finally, the client either requests a shell or execution of a command. The sides then enter session mode. In this mode, either side may send data at any time, and such data is forwarded to/from the shell or command on the server side, and the user terminal on the client side.

When the user program terminates and all forwarded X11 and other connections have been closed, the server sends command exit status to the client, and both sides exit.

sshd can be configured using command-line options or the configuration file /etc/ssh/ssh_config, described in ssh_config(4). Command-line options override values specified in the configuration file.

sshd rereads its configuration file when it receives a hangup signal, SIGHUP, by executing itself with the name it was started as, that is, /usr/lib/ssh/sshd.

Host Access Control

The sshd daemon uses TCP Wrappers to restrict access to hosts. It uses the service name of sshd for hosts_access(). For more information on TCP Wrappers see tcpd(1M) and hosts_access(3) man pages, which are part of the SUNWsfman package (they are not SunOS man pages). TCP wrappers binaries, including libwrap, are in security/tcp-wrapper, a required package for service/network/ssh, the package containing sshd.

オプション

The options for sshd are as follows:

-b *bits*
: Specifies the number of bits in the server key (the default is 768).

-d
: Debug mode. The server sends verbose debug output to the system log, and does not put itself in the background. The server also will not fork and will only process one connection. This option is only intended for debugging for the server. Multiple -d options increase the debugging level. Maximum is 3.

-e
: When this option is specified, sshd will send the output to standard error instead of to the system log.

-f *configuration_file*
: Specifies the name of the configuration file. The default is /etc/ssh/sshd_config. sshd refuses to start if there is no configuration file.

-g *login_grace_time*
: Gives the grace time for clients to authenticate themselves (the default is 300 seconds). If the client fails to authenticate the user within this number of seconds, the server disconnects and exits. A value of zero indicates no limit.

-h *host_key_file*
: Specifies a file from which a host key is read. This option must be given if sshd is not run as root (as the normal host key files are normally not readable by anyone but root). The default is /etc/ssh/ssh_host_key for protocol version 1, and /etc/ssh/ssh_host_rsa_key and /etc/ssh/ssh_host_dsa_key for protocol version 2. It is possible to have multiple host key files for the different protocol versions and host key algorithms.

-h *PKCS#11 URI*
: Instead of working with a host key file, work with a certificate and a private key stored in the PKCS#11 token. See also Using X.509 Certificates section below.

-i
: Specifies that sshd is being run from inetd. sshd is normally not run from inetd because it needs to generate the server key before it can respond to the client, and this may take tens of

seconds. Clients would have to wait too long if the key was regenerated every time. However, with small key sizes (for example, 512) using `sshd` from `inetd` may be reasonable.

`-o` *option*
Can be used to specify options in the format used in the configuration file. This is useful for specifying options for which there are no separate command-line flags.

`-p` *port*
Specifies the port on which the server listens for connections (the default is 22).

`-q`
Quiet mode. Nothing is sent to the system log. Normally the beginning, authentication, and termination of each connection is logged.

`-t`
Test mode. Check only the validity of the configuration file and the sanity of the keys. This is useful for updating sshd reliably as configuration options might change.

`-D`
When this option is specified `sshd` does not detach and does not become a daemon. This allows easy monitoring of `sshd`.

`-4`
Forces `sshd` to use IPv4 addresses only.

`-6`
Forces `sshd` to use IPv6 addresses only.

拡張機能説明

authorized_keys File Format

The `$HOME/.ssh/authorized_keys` file lists the public keys that are permitted for public key authentication (`PubkeyAuthentication`) in protocol version 2. The `AuthorizedKeysFile` configuration option can be used to specify an alternative file.

Each line of the file contains one key (empty lines and lines starting with a hash mark [#] are ignored as comments).

For each RSA key for protocol version 1, the file consists of the following space-separated fields:

options bits exponent modulus comment

For the public key for protocol version 2, the file consists of the following space-separated fields:

options key-type base64-encoding-key comment

For protocol version 2, *key-type* is one of `ssh-rsa` or `ssh-dsa`.

The options field is optional; its presence is determined by whether the line starts with a number. (The option field never starts with a number.) The bits, exponent, and modulus fields give the RSA key; the comment field is a convenient place for you to identify the key.

Lines in this file are usually several hundred bytes long (because of the size of the key modulus). You will find it very inconvenient to type them in; instead, copy the public key file and edit it.

Permissions of this file must be set so that it is not world or group writable. See the `StrictModes` option of `sshd_config(4)`.

The options (if present) consist of comma-separated option specifications. No spaces are permitted, except within double quotes. The following option specifications are supported:

`from="`*pattern-list*`"`
: Specifies that, in addition to public key authentication, the canonical name of the remote host must be present in the comma-separated list of patterns (‘`*`' and ‘`?`' serve as wildcards). The list can also contain negated patterns by prefixing the patterns with ‘`!`'. If the canonical host name matches a negated pattern, the key is not accepted.

  The purpose of this option is to give you the option of increasing security: public key authentication by itself does not trust the network or name servers or anything but the key. However, if someone manages to steal the key, possession of the key would permit the intruder to log in from anywhere in the world. This option makes using a stolen key more difficult, because name servers and routers would have to be compromised, in addition to just the key.

`command="`*command*`"`
: Specifies that the *command* is executed whenever this key is used for authentication. The command supplied by the user (if any) is ignored. The command is run on a `pty` if the client requests a `pty`; otherwise it is run without a `tty`. If an 8-bit clean channel is required, one must not request a `pty` or should specify `no-pty`. You can include a quote in the command by escaping it with a backslash. This option might be useful to restrict certain public keys from performing a specific operation. An example is a key that permits remote backups but nothing else. Note that the client can specify TCP/IP and/or X11 forwarding unless they are explicitly prohibited from doing so. Also note that this option applies to shell, command, or subsystem execution.

`environment="`*NAME=value*`"`
: Specifies that the string *NAME=value* is to be added to the environment when logging in using this key. Environment variables set this way override other default environment values. Multiple options of this type are permitted. Environment processing is disabled by default and is controlled via the `PermitUserEnvironment` option.

`no-port-forwarding`
: Forbids TCP/IP forwarding when this key is used for authentication. Any port forward requests by the client will return an error. This might be used, for example, in connection with the `command` option.

`no-X11-forwarding`
: Forbids X11 forwarding when this key is used for authentication. Any X11 forward requests by the client will return an error.

`no-agent-forwarding`
: Forbids authentication agent forwarding when this key is used for authentication.

`no-pty`
: Prevents `tty` allocation (a request to allocate a `pty` will fail).

`permitopen="`*host*`:`*port*`"`
: Limit local `ssh -L` port forwarding such that it can connect only to the specified host and port. IPv6 addresses can be specified with an alternative syntax: *host*/*port*. You can invoke multiple `permitopen` options, with each instance separated by a comma. No pattern matching is performed on the specified hostnames. They must be literal domains or addresses.

ssh_known_hosts File Format

The `/etc/ssh/ssh_known_hosts` and `$HOME/.ssh/known_hosts` files contain host public keys for all known hosts. The global file should be prepared by the administrator (optional), and the per-user file is maintained automatically: whenever the user connects from an unknown host its key is added to the per-user file.

For the RSA key for protocol version 1, these files consist of the following space-separated fields:

hostnames bits exponent modulus comment

For the public key for protocol version 2, these files consist of the following space-separated fields:

hostnames key-type base64-encoding-key comment

For protocol version 2, *key-type* is one of `ssh-rsa` or `ssh-dsa`.

Hostnames is a comma-separated list of patterns (`*` and `?` act as wildcards); each pattern in turn is matched against the canonical host name (when authenticating a client) or against the user-supplied name (when authenticating a server). A pattern can also be preceded by `!` to indicate negation: if the host name matches a negated pattern, it is not accepted (by that line) even if it matched another pattern on the line.

Alternately, hostnames can be stored in a hashed form, which hides host names and addresses should the file's contents be disclosed. Hashed hostnames start with a vertical bar (`|`) character. Only one hashed hostname can appear on a single line and none of the above negation or wildcard operators may be applied.

Bits, exponent, and modulus are taken directly from the RSA host key; they can be obtained, for example, from `/etc/ssh/ssh_host_rsa_key.pub`. The optional comment field continues to the end of the line, and is not used.

Lines starting with a hash mark (`#`) and empty lines are ignored as comments.

When performing host authentication, authentication is accepted if any matching line has the proper key. It is thus permissible (but not recommended) to have several lines or different host keys for the same names. This will inevitably happen when short forms of host names from different domains are put in the file. It is possible that the files contain conflicting information; authentication is accepted if valid information can be found from either file.

The lines in these files are typically hundreds of characters long. You should definitely not type in the host keys by hand. Rather, generate them by a script or by taking `/etc/ssh/ssh_host_rsa_key.pub` and adding the host names at the front.

環境変数

`sshd` sets the following environment variables for commands executed by `ssh` users:

`DISPLAY`
: Indicates the location of the X11 server. It is automatically set by `sshd` to point to a value of the form *hostname*:*n*, where *hostname* indicates the host where the shell runs, and *n* is an integer greater than or equal to 1. `ssh` uses this special value to forward X11 connections over the secure channel. Unless you have important reasons to do otherwise, you should not set `DISPLAY` explicitly, as that will render the X11 connection insecure and will require you to manually copy any required authorization cookies.

`HOME`
: Set to the path of the user's home directory.

`LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_MONETARY`, `LC_NUMERIC`, `LC_TIME`
: A locale setting. The locale defaults to that of `sshd` (usually the system-wide default locale), or is negotiated between the client and server during initial key exchange (as per RFC 4253).

  Following initial key exchange, each of the variables can be overriden in the following sequence:

  1. If a locale setting is set in a client's environment and that client supports “Environment Variable Passing” (see RFC 4254), then the setting will be passed over to the server side.
  2. If the public key authentication method was used to authenticate the server and the `PermitUserEnvironment` variable in `sshd_config(4)` is set to `yes` on the server side, then the setting can be changed through the use of the `environment` option in the client's `AuthorizedKeysFile` file.
  3. The setting can be change in the client's `~/.ssh/environment` file on the server.

  See `PermitUserEnvironment` in `sshd_config(4)` as to when the `AuthorizedKeysFile` and `~/.ssh/environment` files are processed and used for setting the user environment.

`LOGNAME`
: Synonym for `USER`. Set for compatibility with systems that use this variable.

`MAIL`
: Set to point to the user's mailbox.

SSH_AUTH_SOCK
: Indicates the path of a unix-domain socket used to communicate with the agent.

SSH_CONNECTION
: Identifies the client and server ends of the connection. The variable contains four space-separated values: client IP address, client port number, server IP address and server port number.

SSH_CLIENT
: Identifies the client end of the connection. The variable contains three space-separated values: client IP address, client port number, and server port number.

SSH_TTY
: Set to the name of the tty (path to the device) associated with the current shell or command. If the current session has no tty, this variable is not set.

TZ
: Indicates the present timezone, if TIMEZONE is set in /etc/default/login or if TZ was set when the daemon was started.

HZ
: If set in /etc/default/login, the daemon sets it to the same value.

SHELL
: The user's shell, if ALTSHELL=YES in /etc/default/login.

PATH
: Set to the value of PATH or SUPATH (see login(1)) in /etc/default/login, or, if not set, to /usr/bin:/bin.

USER
: Set to the name of the user logging in.

Additionally, sshd reads $HOME/.ssh/environment and adds lines of the format VARNAME=*value* to the environment.

使用例

In the following examples, certain lines might wrap due to line length limits in your display. You should nevertheless consider the wrapped line as a single line.

例 1 authorized_key File Entries

The following are examples of authorized_key file entries for protocol 1:

```
1024 33 12121...312314325 ylo@foo.bar

from="*.niksula.hut.fi,!pc.niksula.hut.fi" 1024 35 23...2334 ylo@niksula

command="dump /home",no-pty,no-port-forwarding 1024 33 23...2323 backup.hut.fi
```

例 2 authorized_key File Entries for Protocol 2

The following are examples of `authorized_key` file entries for protocol 2:

```
ssh-rsa AAAAB3NzaC1y.....EU88ovYKg4GfclWGCFYTuw8= ylo@foo.bar
from="*.niksula.hut.fi" ssh-rsa AAAAB3NzaC...uw8= ylo@niksula
command="dump /home",no-pty,no-port-forwarding ssh-rsa AA..8= backup.hut.fi
```

例 3 ssh_known_hosts File Entries for Protocol 1

The following are examples of `ssh_known_hosts` file entries for protocol 1:

```
closenet,closenet.hut.fi,...,130.233.208.41 1024 37 159...93 closenet.hut.fi
```

例 4 ssh_known_hosts File Entries for Protocol 2

The following are examples of `ssh_known_hosts` file entries for protocol 2:

```
closenet,closenet.hut.fi,...,130.233.208.41 ssh-rsa AA..8= closenet.hut.fi
```

例 5 Using X.509 Public Key Authentication

The following example of user authentication should help you understand how the X.509 public key authentication works. Steps to use X.509 certificates/keys for host authentication are very similar. Note that we use the same token (`softtoken`) for both the TA certificate and the user certificate, which would not normally be the case.

Create a self-signed trusted anchor certificate and a user certificate signed by the trusted anchor private key. Configure the SSH daemon for certificate validation, and run the SSH client with the user certificate generated in the previous step as its user identity.

Replace user name "*PUT-YOUR-USERNAME-HERE*" with an existing (your) user name on the server side.

```
# Generate a trusted anchor certificate/key pair in the
# default token (softtoken).
pktool gencert keystore=pkcs11 label=authority \
    subject="CN=authority" serial=0x01

# Export the trusted anchor certificate (and, after that,
# put it to /etc/ssh/cert directory).
pktool export keystore=pkcs11 outfile=ta.cert objtype=cert
    label=authority outformat=pem

# Generate a user certificate signed by the trusted anchor
# private key and then import the certificate into the
# token.
pktool gencsr keystore=pkcs11 label=user \
    outcsr=user.csr subject="CN=<PUT-YOUR-USERNAME-HERE>"
pktool signcsr keystore=pkcs11 signkey=authority \
    csr=user.csr serial=0x02 issuer="CN=authority" \
```

例 5 Using X.509 Public Key Authentication (続き)

```
     outcert=user.cert
pktool import keystore=pkcs11 infile=user.cert label=user

# Create a new policy file.
kmfcfg create dbfile=/etc/ssh/policy.xml policy=ssh \
    ta-name=search mapper-name=cn

# Start the SSH daemon in a debug mode to test the
# configuration. Use a different port so that you do not
# clash with the already running daemon.
/usr/lib/ssh/sshd -p 2222 -ddd -o \
    -o TrustedAnchorKeystore=/etc/ssh/cert \
    -o KMFPolicyDatabase=/etc/ssh/policy.xml \
    -o KMFPolicyName=ssh

# Run the client in a debug mode to test the configuration
# (will ask for a PIN for the token, use the 'pinfile'
# attribute if you do not want to set PIN for each
# invocation).
ssh -p 2222 -vvv -o \
    IdentityFile=\
    "pkcs11:object=user;token=Sun Software PKCS#11 softtoken" \
    <PUT-YOUR-USERNAME-HERE>@localhost
```

Before you use the softtoken you should set its PIN by means of the `setpin` subcommand. See `pktool(1)` for more information.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

ファイル

`/etc/default/login`

Contains defaults for several `sshd_config` parameters, environment variables, and other environmental factors.

The following parameters affect environment variables (see `login(1)` and descriptions of these variables, above):

- `TIMEZONE`
- `HZ`
- `ALTSHELL`
- `PATH`
- `SUPATH`

The following `/etc/default/login` parameters supply default values for corresponding `sshd_config(4)` parameters:

- CONSOLE (see PermitRootLogin in sshd_config(4))
- PASSREQ (see PermitEmptyPasswords in sshd_config(4))
- TIMEOUT (see LoginGraceTime in sshd_config(4))

The following /etc/default/login parameters:

- UMASK
- ULIMIT

...set the umask(2) and file size limit of, respectively, the shells and commands spawned by sshd.

Finally, two /etc/default/login parameters affect the maximum allowed login attempts per-connection using interactive user authentication methods (for example, keyboard-interactive but not publickey), as per login(1):

- RETRIES
- SYSLOG_FAILED_LOGINS

/etc/ssh/sshd_config
: Contains configuration data for sshd. This file should be writable by root only, but it is recommended (though not necessary) that it be world-readable.

/etc/ssh/ssh_host_key
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_rsa_key
: Contains the private part of the host key. This file should only be owned by root, readable only by root, and not accessible to others. sshd does not start if this file is group/world-accessible.

/etc/ssh/ssh_host_key.pub
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_rsa_key.pub
: Contains the public part of the host key. This file should be world-readable but writable only by root. Its contents should match the private part. This file is not used for encryption; it is provided only for the convenience of the user so its contents can be copied to known hosts files. These two files are created using ssh-keygen(1).

/var/run/sshd.pid
: Contains the process ID of the sshd listening for connections. If there are several daemons running concurrently for different ports, this contains the pid of the one started last. The content of this file is not sensitive; it can be world-readable. You can use the PidFile keyword in sshd_config to specify a file other than /var/run/sshd.pid. See sshd_config(4).

/etc/ssh/ssh_known_hosts and $HOME/.ssh/known_hosts
: These files are consulted when using rhosts with public key host authentication to check the public key of the host. The key must be listed in one of these files to be accepted. The client uses the same files to verify that the remote host is the one it intended to connect.

These files should be writable only by root or the owner. `/etc/ssh/ssh_known_hosts` should be world-readable, and `$HOME/.ssh/known_hosts` can but need not be world-readable.

`/etc/nologin`
If this file exists, `sshd` refuses to let anyone except root log in. The contents of the file are displayed to anyone trying to log in, and non-root connections are refused. The file should be world-readable.

`$HOME/.ssh/authorized_keys`
Lists the public keys (RSA or DSA) that can be used to log into the user's account. This file must be readable by root. This might, on some machines, imply that it is world-readable if the user's home directory resides on an NFS volume. It is recommended that it not be accessible by others. The format of this file is described above. Users will place the contents of their `identity.pub`, `id_dsa.pub` and/or `id_rsa.pub` files into this file, as described in `ssh-keygen(1)`.

`$HOME/.rhosts`
This file contains host-username pairs, separated by a space, one per line. The given user on the corresponding host is permitted to log in without password. The same file is used by `rlogind` and `rshd`. The file must be writable only by the user; it is recommended that it not be accessible by others. It is also possible to use `netgroups` in the file. Either host or user name may be of the form +@*groupname* to specify all hosts or all users in the group.

`$HOME/.shosts`
For `ssh`, this file is exactly the same as for `.rhosts`. However, this file is not used by `rlogin` and `rshd`, so using this permits access using SSH only.

`/etc/hosts.equiv`
This file is used during `.rhosts` authentication. In its simplest form, this file contains host names, one per line. Users on these hosts are permitted to log in without a password, provided they have the same user name on both machines. The host name can also be followed by a user name; such users are permitted to log in as any user on this machine (except root). Additionally, the syntax +@*group* can be used to specify netgroups. Negated entries start with a hyphen (`-`).

If the client host/user is successfully matched in this file, login is automatically permitted, provided the client and server user names are the same. Additionally, successful RSA host authentication is normally required. This file must be writable only by root; it is recommended that it be world-readable.

Warning: It is almost never a good idea to use user names in `hosts.equiv`. Beware that it really means that the named user(s) can log in as anybody, which includes `bin`, `daemon`, `adm`, and other accounts that own critical binaries and directories. For practical purposes, using a user name grants the user root access. Probably the only valid use for user names is in negative entries. This warning also applies to `rsh`/`rlogin`.

`/etc/ssh/moduli`
A private file.

`/etc/ssh/shosts.equiv`
This file is processed exactly as `/etc/hosts.equiv`. However, this file might be useful in environments that want to run both `rsh`/`rlogin` and `ssh`.

`$HOME/.ssh/environment`
This file is read into the environment at login (if it exists). It can contain only empty lines, comment lines (that start with #), and assignment lines of the form *name*=*value*. The file should be writable only by the user; it need not be readable by anyone else. Environment processing is disabled by default and is controlled by means of the `PermitUserEnvironment` option.

`$HOME/.ssh/rc`
If this file exists, it is run with `/bin/sh` after reading the environment files but before starting the user's shell or command. If X11 spoofing is in use, this will receive the `proto cookie` pair in standard input (and `DISPLAY` in environment). This must call `xauth` in that case.

The primary purpose of `$HOME/.ssh/rc` is to run any initialization routines that might be needed before the user's home directory becomes accessible; AFS is a particular example of such an environment. If this file exists, it is run with `/bin/sh` after reading the environment files, but before starting the user's shell or command. It must not produce any output on stdout; stderr must be used instead. If X11 forwarding is in use, it will receive the `proto cookie` pair in its standard input and `DISPLAY` in its environment. The script must call `xauth` because `sshd` will not run `xauth` automatically to add X11 cookies.

This file will probably contain some initialization code followed by something similar to:

```
if read proto cookie && [ -n "$DISPLAY" ]
then
  if [ ‘echo $DISPLAY | cut -c1-10‘  =  ’localhost:’ ]
  then
    # X11UseLocalhost=yes
    echo add unix:‘echo $DISPLAY |
    cut -c11-‘ $proto $cookie
  else
    # X11UseLocalhost=no
    echo add $DISPLAY $proto $cookie
  fi | xauth -q -
fi
```

If this file does not exist, `/etc/ssh/sshrc` is run, and if that does not exist, `xauth` is used to store the cookie. `$HOME/.ssh/rc` should be writable only by the user, and need not be readable by anyone else.

/etc/ssh/sshrc
: Similar to $HOME/.ssh/rc. This can be used to specify machine-specific login-time initializations globally. This file should be writable only by root, and should be world-readable.

セキュリティー

sshd supports the use of several user authentication mechanisms: a public key system where keys are associated with users (through users' authorized_keys files), a public key system where keys are associated with hosts (see the HostbasedAuthentication configuration parameter), a GSS-API based method (see the GssAuthentication and GssKeyEx configuration parameters) and three initial authentication methods: none, password, and a generic prompt/reply protocol, keyboard-interactive.

sshd negotiates the use of the GSS-API with clients only if it has a GSS-API acceptor credential for the "host" service. This means that, for GSS-API based authentication, the server must have a Kerberos V keytab entry (see below) or the equivalent for any other GSS-API mechanism that might be installed.

In order for Kerberos authentication to work, a host/<*FQDN*> Kerberos principal must exist for each Fully Qualified Domain Name associated with the in.sshd server. Each of these host/<*FQDN*> principals must have a keytab entry in the /etc/krb5/krb5.keytab file on the in.sshd server. An example principal might be:

```
host/bigmachine.eng.example.com
```

See kadmin(1M) or gkadmin(1M) for instructions on adding a principal to a krb5.keytab file. See 『Oracle Solaris の管理: セキュリティーサービス』 for a discussion of Kerberos authentication.

GSS-API authorization is covered in gss_auth_rules(5).

sshd uses pam(3PAM) for the three initial authentication methods as well as for account management, session management, and password management for all authentication methods.

Specifically, sshd calls pam_authenticate() for the "none," "password" and "keyboard-interactive" SSHv2 userauth types. Other SSHv2 authentication methods do not call pam_authenticate(). pam_acct_mgmt() is called for each authentication method that succeeds.

pam_setcred() and pam_open_session() are called when authentication succeeds and pam_close_session() is called when connections are closed.

pam_open_session() and pam_close_session() are also called when SSHv2 channels with ptys are opened and closed.

Each SSHv2 userauth type has its own PAM service name:

SSHv2 Userauth	PAM Service Name
none	sshd-none
password	sshd-password
keyboard-interactive	sshd-kbdint
pubkey	sshd-pubkey
hostbased	sshd-hostbased
gssapi-with-mic	sshd-gssapi
gssapi-keyex	sshd-gssapi

If `pam_acct_mgmt()` returns `PAM_NEW_AUTHTOK_REQD` (indicating that the user's authentication tokens have expired), then `sshd` forces the use of "keyboard-interactive" `userauth`, if version 2 of the protocol is in use. The "keyboard-interactive" `userauth` will call `pam_chauthtok()` if `pam_acct_mgmt()` once again returns `PAM_NEW_AUTHTOK_REQD`. By this means, administrators are able to control what authentication methods are allowed for SSHv2 on a per-user basis.

Setting up Host-based Authentication

To establish host-based authentication, you must perform the following steps:

- Configure the client.
- Configure the server.
- Publish known hosts.
- Make appropriate entries in `/etc/ssh/shosts.equiv` and `~/.shosts`.

These steps are expanded in the following paragraphs.

- On a client machine, in the system-wide client configuration file, `/etc/ssh/ssh_config`, you must have the entry:

  ```
  HostbasedAuthentication yes
  ```

  See `ssh_config(4)` and `ssh-keysign(1M)`.
- On the server, in the system-wide server configuration file, `/etc/ssh/sshd_config`, you must have the entry:

  ```
  HostbasedAuthentication yes
  ```

  If per-user `.shost` files are to be allowed (see last step), in the same file, you must have:

  ```
  IgnoreRhosts no
  ```

  See `sshd_config(4)` for a description of these keywords.
- To publish known hosts, you must have entries for the clients from which users will be allowed host-based authentication. Make these entries in either or both of the system-wide file (`/etc/ssh/ssh_known_hosts`) or the per-user file (`~/.ssh/known_hosts`).

- Note that sshd uses .shosts, not .rhosts. If you want the functionality provided by .rhosts, but do not want to use rlogin or rsh because of their security shortcomings, you can use .shosts in conjunction with sshd. To use this feature, make appropriate entries in /etc/ssh/shosts.equiv and ~/.shosts, in the format specified in rhosts(4).

  For the vast majority of network environments, .shosts is preferred over .rhosts.

Using X.509 Certificates as Host Keys and User Identities

SSH commands can work with X.509v3 certificates when used as host keys and/or user identities. When used this way, PKCS#11 URIs can be used instead of plain filenames to locate keys and certificates. Host keys, user identities, and their corresponding certificates used for X.509 public key authentication can be stored only in PKCS#11 tokens. The only PKCS#11 URI attributes used by SSH are object, token, and pinfile. If no hardware keystore is available, the PKCS#11 softtoken keystore can be used.

For certificate validation (a server that authenticates a user, or a client that authenticates a server), the KMF policy needs a certificate mapper. Mappers are not set in the default policy file. See TrustedAnchorKeystore, KMFPolicyDatabase, and KMFPolicyName options in sshd_config(4), and see libkmf(3LIB) for more information.

Self-signed certificates can only be used as host keys, never as user identities. If a self-signed certificate is used or if a host certificate cannot be validated on the client side due to a missing relevant trusted anchor certificate, the SSH client offers to store the key in the known_host database, as is currently done with plain public keys. Only the public key is extracted from the certificate and stored in the database, thus maintaining the current format. The reason for accepting such a certificate on the client side is that the initial host Key Exchange is not restartable. Given this, if a server uses certificates as its only host key(s), a client that is unable to validate the certificate would otherwise be denied from logging in.

See the EXAMPLES section for guidance on how to use the X.509 authentication.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/ssh
Interface Stability	Committed

The interface stability of /etc/ssh/moduli is Private.

関連項目

kmfcfg(1), login(1), pktool(1), scp(1), ssh(1), ssh-add(1), ssh-agent(1), ssh-keygen(1), svcs(1), gkadmin(1M), kadmin(1M), sftp-server(1M), ssh-keysign(1M), svcadm(1M), libkmf(3LIB), pam(3PAM), rhosts(4), ssh_config(4), sshd_config(4), attributes(5), gss_auth_rules(5), kerberos(5), pam_roles(5), pkcs11_softtoken(5), smf(5)

『Oracle Solaris の管理: セキュリティーサービス』

注意事項

The sshd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/ssh:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

sshd always sets PAM_RHOST and sets PAM_AUSER in the case of host-based userauth. This behavior allows for remote logins to roles using host-based authentication. See pam_roles(5).

名前

ssh-keysign – ssh helper program for host-based authentication

形式

`ssh-keysign`

機能説明

`ssh-keysign` is used by `ssh(1)` to access the local host keys and generate the digital signature required during host-based authentication with SSH protocol version 2. This signature is of data that includes, among other items, the name of the client host and the name of the client user.

`ssh-keysign` is disabled by default and can be enabled only in the global client configuration file `/etc/ssh/ssh_config` by setting `HostbasedAuthentication` to `yes`.

`ssh-keysign` is not intended to be invoked by the user, but from `ssh`. See `ssh(1)` and `sshd(1M)` for more information about host-based authentication.

ファイル

`/etc/ssh/ssh_config`	Controls whether `ssh-keysign` is enabled.
`/etc/ssh/ssh_host_dsa_key` `/etc/ssh/ssh_host_rsa_key`	These files contain the private parts of the host keys used to generate the digital signature. They should be owned by root, readable only by root, and not accessible to others. Because they are readable only by root, `ssh-keysign` must be `set-uid` root if host-based authentication is used.

セキュリティー

ssh-keysign will not sign host-based authentication data under the following conditions:

- If the `HostbasedAuthentication` client configuration parameter is not set to `yes` in `/etc/ssh/ssh_config`. This setting cannot be overriden in users' `~/.ssh/ssh_config` files.
- If the client hostname and username in `/etc/ssh/ssh_config` do not match the canonical hostname of the client where `ssh-keysign` is invoked and the name of the user invoking `ssh-keysign`.

In spite of `ssh-keysign`'s restrictions on the contents of the host-based authentication data, there remains the ability of users to use it as an avenue for obtaining the client's private host keys. For this reason host-based authentication is turned off by default.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ssh
Interface Stability	Committed

関連項目

`ssh(1)`, `sshd(1M)`, `ssh_config(4)`, `attributes(5)`

Authors Markus Friedl, `markus@openbsd.org`

History `ssh-keysign` first appeared in Ox 3.2.

名前

statd – network status monitor

形式

/usr/lib/nfs/statd

機能説明

statd is an intermediate version of the status monitor. It interacts with lockd(1M) to provide the crash and recovery functions for the locking services on NFS. statd keeps track of the clients with processes which hold locks on a server. When the server reboots after a crash, statd sends a message to the statd on each client indicating that the server has rebooted. The client statd processes then inform the lockd on the client that the server has rebooted. The client lockd then attempts to reclaim the lock(s) from the server.

statd on the client host also informs the statd on the server(s) holding locks for the client when the client has rebooted. In this case, the statd on the server informs its lockd that all locks held by the rebooting client should be released, allowing other processes to lock those files.

lockd is started by automountd(1M), mount_nfs(1M), and share(1M) if NFS automounts are needed.

ファイル

/var/statmon/sm	lists hosts and network addresses to be contacted after a reboot
/var/statmon/sm.bak	lists hosts and network addresses that could not be contacted after last reboot
/var/statmon/state	includes a number which changes during a reboot
/usr/include/rpcsvc/sm_inter.x	contains the rpcgen source code for the interface services provided by the statd daemon.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目

svcs(1), automountd(1M), lockd(1M), mount_nfs(1M), share(1M), svcadm(1M), attributes(5), smf(5)

『Oracle Solaris の管理: IP サービス』

注意事項

The crash of a server is only detected upon its recovery.

The statd service is managed by the service management facility, smf(5), under the service identifier:

svc:/network/nfs/status

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

If it is disabled, it will be enabled by mount_nfs(1M), share_nfs(1M), and automountd(1M) unless its application/auto_enable property is set to false.

名前

stclient – Service Tag Administration Program

形式

```
stclient -x | -E [-r root_dir]

stclient -a [-i instance_URN] -p product_name -e product_version
     -t product_URN [-F parent_URN] -P product_parent
     [-I product_defined_instance_id] -m product_vendor -A platform_arch
     -z container -S source [-r root_dir]

stclient -u -i instance_URN -F parent_URN -I product_defined_instance_id
     [-r root_dir]

stclient -d -i instance_URN [-r root_dir]

stclient -g -i instance_URN [-r root_dir]

stclient -f -t product_URN [-r root_dir]

stclient -h

stclient -v
```

機能説明

The `stclient` command displays, finds, adds, updates and deletes records in the Service Tag registry. The registry is in the XML file `/var/sadm/servicetag/registry/servicetag.xml`, and contains the inventory of the product instances installed in the machine. Each record has a unique product instance identifier which is generated when the service tag is added in the registry. This product instance identifier is used as a key when finding, updating or deleting the service tag records. The extract option prints out the registry contents in XML format in stdout.

The `stclient` command also runs in interactive mode. This mode is invoked by running `stclient` without any parameters. A menu of all the available options are displayed, and the user is prompted to enter different parameters depending on the option chosen.

Any user can extract or get the contents of the registry, but only users with the appropriate privileges, the “svctag” user, or the creator of the service tag is authorized to update or delete a service tag record.

オプション

The following options are supported:

`-p` *product_name*	Sets the product name of the service tag to be added. For example, “stclient -p “Solaris 10 Operating System”” sets the product name to the “Solaris 10 Operating System”.
`-e` *product_version*	Sets the product version of the service tag to be added. For example, “stclient -e 5.10” sets the product version to “5.10”.
`-t` *product_URN*	Sets the Sun product unique identifier of the service tag to be added. For example, “stclient -t urn:uuid:5005588c-36f3-11d6-9cec-fc96f718e113” sets

	the Sun product unique identifier to “urn:uuid:5005588c-36f3-11d6-9cec-fc96f718e113”.
`-i` *instance_URN*	Sets the product instance unique identifier of the service tag. For example, “stclient -i 3a444ab2-1dd2-11b2-a69d-830020782a6b” sets the product instance unique identifier to “3a444ab2-1dd2-11b2-a69d-830020782a6b” This field may also be set when the service tag is added but is not typically used. The generation of the instance ID is normally left to the `stclient`.
`-F` *product_parent_URN*	Sets the Sun product unique identifier of parent product. For example, “stclient -F urn:uuid:5005588c-36f3-11d6-9cec-fc96f718e113” sets the Sun product unique identifier of parent product to “urn:uuid:5005588c-36f3-11d6-9cec-fc96f718e113”.
`-P` *product_parent*	Sets the parent product of the service tag. For example, “stclient -P JES” sets the product parent to “JES”.
`-I` *product_defined_instance_id*	Sets the product defined instance identifier. For example, “stclient -I ser:abc-klm-000-7190” sets the product defined instance identifier to “ser:abc-klm-000-7190”.
`-m` *product_vendor*	Sets the vendor of the product. For example, “stclient –m Sun” sets the product vendor to “Sun”.
`-A` *platform_arch*	Sets the platform architecture that the product is running on. For example, “stclient –A sparc” sets the platform architecture to “sparc”
`-z` *container*	Sets the container that the product is running on. For example, “stclient -z “global zone”” sets the container to “global zone”.
`-S` *source*	Sets the source of this service tag, naming the entity that created it.
`-r` *root_dir*	Sets the root directory where the command searches for the registry. The command searches `/var/sadm/servicetag/registry` by default. When this option is used, the command looks for the Registry in the specified root directory, (for example, “`-r /home/user1`” searches `/home/user1/var/sadm/servicetag`). This is typically used for testing.

オペランド

Function Letters

The function portion of the key is specified by one of the following letters:

x Extract. Extracts and prints the contents of the Service Tag registry in XML format. An alternate root directory may be specified for testing purposes.

a Add. Adds a service tag in the registry. A unique product instance identifier may be supplied and is automatically generated if not provided. This field is returned by the command. The required fields for add are product name, product version, product URN, product parent, vendor, platform architecture, container and source.

u Update. Updates a service tag in the registry using the product instance URN as the key. The parent URN and product defined instance id fields can be updated.

d Delete. Deletes a service tag in the registry using the product instance URN as the key.

E Extract Environment. Enumerates to standard output the environmental or “agent”' information associated with the registered Service Tags on this system.

g Get. Gets and prints a service tag from the registry using the product instance URN as the key.

f Find. Finds and prints all the instance URNs of a given product URN.

h Help. Displays the command options.

v Version. Displays the version number of the command.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`> 0` An error occurred.

使用例

例 1 Adding a Service Tag in the Registry

To add a service tag, enter the following:

```
# stclient -a -p "Java Enterprise Web Server 6.0" -e 6.0 \
   -t urn:uuid:f2b8b59c-6eba-11d7-986f-9f5d47ec72fe  \
   -P Java Enterprise Server -m Sun -A sparc -z global -S patch
```

The screen displays the following:

```
Java Enterprise Web Server 6.0 6.0 added
Product instance URN=urn:st:7fc61914-1dd2-11b2-b992-830020782a6b
```

例 2 Updating a Service Tag in the Registry

To update a service tag, enter the following:

例 2 Updating a Service Tag in the Registry *(続き)*

```
# stclient -u -i 7fc61914-1dd2-11b2-b992-830020782a6b \
   -I urn:st:product.defined.id
```

The screen displays the following:

```
Service tag updated
```

例 3 Extracting a Service Tag Registry

To extract a service tag, enter the following:

```
# stclient -x
```

The screen displays output similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<registry urn="urn:uuid:1234ab-00e1-11b3-98737646873" version="1.0">
   <service_tag>
  .
  .
  .
   </service_tag>
 </registry>
```

例 4 Finding all product instances

To find all product instances, enter the following:

```
# stclient -f -t urn:uuid:f2b8b59c-6eba-11d7-986f-9f5d47ec72fe \
   fc61914-1dd2-11b2-b992-830020782a6b
```

例 5 Listing the Environmental Information

To list the environmental information associated with the registered Service Tags on this system, enter the following:

```
# stclient -E
```

The screen displays output similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<agent>
   <agent_urn>urn:st:af15ee62-0bb3-ef2d-fa96-85a11996cc71</agent_urn>
  .
  .
  .
   </system_info>
 </agent>
```

環境変数

See environ(5)for descriptions of the following environment variables that affect the execution of stclient: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/service-tag
Interface Stability	Committed

関連項目

in.stdiscover(1M), in.stlisten(1M)

名前

stmfadm – SCSI ターゲットモードフレームワーク (SCSI Target Mode Framework、SCSI) のコマンド行インタフェース

形式

```
stmfadm add-hg-member -g, --group-name group-name group-member...

stmfadm add-tg-member -g, --group-name group-name group-member...

stmfadm add-view [-n, --lun logical-unit-number
     -t, --target-group group-name -h, --host-group group-name] lu-name

stmfadm create-hg group-name

stmfadm create-lu [-p, --lu-prop logical-unit-property=val
     -s, --size size] lu-file

stmfadm create-tg group-name

stmfadm delete-hg group-name

stmfadm delete-lu lu-name

stmfadm delete-tg group-name

stmfadm import-lu lu-file

stmfadm list-hg [-v] [host-group-name...]

stmfadm list-tg [-v] [target-group-name...]

stmfadm list-lu [-v] [lu-name...]

stmfadm list-target [-v] [target-name...]

stmfadm list-view -l, --lu-name lu-name [entry-name...]

stmfadm list-state

stmfadm modify-lu [-p, --lu-prop logical-unit-property=val
     -s, --size size, -f, --file] lu-name|lu-file

stmfadm online-lu lu-name

stmfadm offline-lu lu-name

stmfadm online-lu target-name

stmfadm offline-lu target-name

stmfadm remove-hg-member -g, --group-name group-name group-member...

stmfadm remove-tg-member -g, --group-name group-name group-member...

stmfadm remove-view -l, --lu-name lu-name entry-name
```

機能説明

stmfadm コマンドは、STMF フレームワーク内部の論理ユニットを構成します。このフレームワークおよびこのマニュアルページでは、次の用語を使用します。

イニシエータ
SCSI I/O コマンドを SCSI ターゲットと論理ユニットに発行する役割を果たすデバイス。

ターゲット
論理ユニットへの SCSI I/O コマンドを受信する役割を果たすデバイス。

論理ユニット
ターゲット内にあり、SCSI I/O コマンドを実行する役割を果たすデバイス。

論理ユニット番号
ターゲット内の論理ユニットの識別子。

イニシエータグループ
イニシエータグループは、ビュー (下記を参照) に適用するために結合された 1 つ以上のイニシエータのセットです。1 つのイニシエータが複数のイニシエータグループのメンバーにはなることはできません。

ターゲットグループ
ターゲットグループは、ビュー (下記を参照) を作成するときに同じように扱われる 1 つ以上の SCSI ターゲットポートのセットです。特定の SCSI イニシエータが認識できる論理ユニットのセットは、ビューの結合されたセットによって決まります。

各論理ユニットにはビューエントリのセットが含まれ、各ビューエントリは、ターゲットグループ、ホストグループ、および LUN を指定します。そのホストグループからのイニシエータがそのターゲットグループを介して接続する場合、イニシエータはその論理ユニットを識別し、指定された LUN を使用してその論理ユニットに接続できます。ビューを使用すると、特定のイニシエータが認識できる論理ユニットのセットを制限したり、使用される LUN のセットを割り当てたりできます。

ビュー
ビューは、イニシエータグループ、ターゲットグループ、および論理ユニット番号と、指定された論理ユニットとの関連付けを定義します。論理ユニットに追加されるどのビューエントリも、その論理ユニットの既存のビューエントリと競合してはいけません。ビューエントリが競合していると見なされるのは、任意のイニシエータ、ターゲット、および論理ユニット番号の関連付けの複製が試みられた場合です。たとえば、論理ユニット `LU_0` に、次のビューエントリが関連付けられているとします。

```
Logical Unit: LU_0
            View Entry: 0
                initiator group: HostA
                target group: All targets
                logical unit number: 32
```

次を試みた場合、

```
# stmf add-view -n 31 -h HostA LU_0
```

この操作によってエラーが返され、このビューエントリが 1 つ以上の既存のビューエントリと競合していることを示すメッセージが表示されます。この競合は、既存のビューエントリ 0 がすでに LU_0 をイニシエータグループ HostA の論理ユニット番号 32 にマップしているために発生します。

サブコマンド

stmfadm コマンドは、次のサブコマンドをサポートします。

add-view [-n, --lun *logical-unit-number* -t, --target-group *group-name* -h, --host-group *group-name*] *lu-name*

論理ユニットのビューエントリを論理ユニット *lu-name* に追加します。ここで、*lu-name* は、list-lu サブコマンドで表示される論理ユニットの STMF 名です。add-view サブコマンドは、論理ユニットのアクセス制御を実装するためのメカニズムをユーザーに提供するだけでなく、イニシエータとターゲットの特定のセットの論理ユニットに論理ユニット番号を割り当てるための手段も提供します。論理ユニットは、少なくとも 1 つのビューが適用されるまでは、どのイニシエータからも使用できません。各ビューエントリにはエントリ名が割り当てられます。このエントリ名を使用すると、remove-view サブコマンドや list-view サブコマンドでエントリを参照できます。

add-view は次のオプションをサポートします。

-n, --lun *logical-unit-number*

logical-unit-number は、このビューエントリの論理ユニットに割り当てられる 0 - 16383 の範囲の整数です。このオプションが指定されていない場合、論理ユニット番号は STMF フレームワークによって割り当てられます。

-t, --target-group *group-name*

group-name は、STMF の create-tg サブコマンドを使用して以前に作成されたターゲットグループの名前です。このオプションが指定されていない場合、この論理ユニットはすべてのターゲットから使用できます。

-h, -host-group *group-name*

group-name は、STMF の create-hg サブコマンドを使用して以前に作成されたホストグループの名前です。このオプションが指定されていない場合、この論理ユニットは、STMF フレームワークにログインするすべてのイニシエータから使用できます。

add-hg-member -g *group-name group member...*

ホストグループメンバーをホストグループに追加します。*group-name* は、create-hg サブコマンドを使用して作成された既存のグループである必要があります。*group member* は、*name_type.name_value* として指定できます。ここで、*name_type* には次のいずれかを指定できます。

wwn
iqn
eui

また、*name_value* はイニシエータ名の値です。たとえば、ファイバチャネルイニシエータポートのワールドワイドネーム 200000e08b909221 をホストグループ HostA に追加するには、次のコマンドを入力します。

```
# stmfadm add-hg-member -g HostA wwn.200000e08b909221
```

iqn.1986-03.com.sun:01.46f7e262 という名前の ISCSI イニシエータノードメンバーを HostA に追加するには、次のコマンドを入力します。

```
# stmfadm add-hg-member -g HostA iqn.1986-03.com.sun:01.46f7e262
```

あるいは、メンバーをそれぞれの SCSI 名文字列識別子で指定することもできます。上の 2 つのイニシエータをそれぞれの SCSI 名文字列識別子を使用して追加するには、次のコマンドを入力します。

```
# stmfadm add-hg-member -g HostA eui.200000e08b909221
# stmfadm add-hg-member -g HostA iqn.1986-03.com.sun:01.46f7e262
```

1 つのホストグループメンバーが複数のホストグループのメンバーになることはできません。

add-tg-member -g *group-name group member...*

ターゲットグループメンバーをターゲットグループに追加します。*group-name* は、create-tg サブコマンドを使用して作成された既存のグループである必要があります。*group member* は、*name_type.name_value* として指定できます。ここで、*name_type* には次のいずれかを指定できます。

```
wwn
iqn
eui
```

また、*name_value* はターゲット名の値です。たとえば、ファイバチャネルターゲットポートのワールドワイドネーム 501000e092376af7 をターゲットグループ TG0 に追加するには、次のコマンドを入力します。

```
# stmfadm add-tg-member -g TG0 wwn.501000e092376af7
```

iqn.1986-03.com.sun:target.01.01110 という名前の ISCSI ターゲットメンバーを TG0 に追加するには、次のコマンドを入力します。

```
# stmfadm add-tg-member -g TG0 iqn.1986-03.com.sun:target.01.01110
```

あるいは、メンバーをそれぞれの SCSI 名文字列識別子で指定することもできます。上の 2 つのターゲットをそれぞれの SCSI 名文字列識別子を使用して追加するには、次のコマンドを入力します。

```
# stmfadm add-tg-member -g TG0 eui.501000e092376af7
# stmfadm add-tg-member -g TG0 iqn.1986-03.com.sun:target.01.01110
```

1 つのターゲットグループメンバーが複数のターゲットグループのメンバーになることはできません。

`create-hg` *group-name*
group-name という名前のイニシエータグループを作成します。*group-name* は、最大長が 255 の Unicode 文字の文字列です。このグループ名は、STMF システム内で一意である必要があります。

`create-lu` [`-p`, `--lu-prop` *logical-unit-property=val* `--s`, `--size` *size*] *lu-file*
STMF に登録できる論理ユニットを作成します。`-p` オプションでは、*logical-unit-property* に次のいずれかのプロパティーを指定できます。

`alias`
最大 255 文字。デバイスのユーザー定義の名前を表します。デフォルトはバッキングストアの名前です。

`blk`
デバイスのブロックサイズを指定します。デフォルトは 512 ですが、この値を 4096K バイトより高い値に設定しないでください。

`guid`
有効な NAA Registered Extended Identifier を表す 32 個の 16 進 ASCII 文字。デフォルトは、STMF によって設定された生成値になります。

`host-id`
ホスト ID の割り当てを表す 8 個の 16 進 ASCII 文字。これは、論理ユニットのグローバル一意識別子 (GUID) を生成するために使用されます。デフォルトは `hostid(1)` によって返される値です。

`meta`
メタデータファイル名。指定されている場合は、論理ユニットの SCSI メタデータを保持するために使用されます。シグナルを送るプロセスまたはジョブにデフォルトはありません。

`mgmt-url`
最大 1024 文字。管理ネットワークアドレスの URL を表します。一対の単一引用符 (') で囲まれたスペース区切りの URL を使用すると、複数の URL を 1 つのパラメータとして渡すことができます。

`oui`
組織の一意の識別子。IEEE OUI 企業 ID の割り当てを表す 6 個の 16 進 ASCII 文字。これは、デバイス識別子 (GUID) を生成するために使用されます。デフォルトは `00144F` です。

`pid`
製品 ID の SCSI SPC-3 を表す 16 個の文字。この値は、デバイス用に返される標準の `INQUIRY` データに反映されます。デフォルトは `COMSTAR` です。

`serial`
シリアル番号。SCSI の重要な製品データのシリアル番号 (ページ `80h`) を指定します。これは、最大 252 バイト長のキャラクタ値です。デフォルト値はありません。

`vid`
SCSI SPC-3 ごとのベンダー ID を表す 8 個の文字。この値は、デバイス用に返される標準の `INQUIRY` データに反映されます。デフォルトは `SUN` です。

`wcd`
ライトバックキャッシュ無効。ライトバックキャッシュ無効の動作を決定するには、`true` または `false` を指定します。デフォルトは、*lu-file* 引数で指定されたバッキングストアデバイスのライトバックキャッシュ設定です。

`wp`
書き込み保護ビット。書き込み保護されているデバイスとして報告するかどうかを決定するには、`true` または `false` を指定します。デフォルトは `false` です。

`-s` オプションでは、*size* として、整数のあとにサイズの単位を示す次のいずれかの文字を指定します。

`k` K バイト

`m` M バイト

`g` G バイト

`t` T バイト

`p` P バイト

`e` E バイト

lu-file は、論理ユニットのバッキングストアとして使用されるファイルです。`-s` オプションが指定されていない場合は、指定された *lu-file* のサイズが論理ユニットのサイズとして使用されます。STMF に登録された論理ユニットには、メタデータを格納するための領域が必要です。バッキングストアデバイスとして `zvol` が指定されている場合、デフォルトでは、`zvol` の特殊なプロパティーを使用してメタデータが格納されます。そのほかのすべてのデバイスでは、デフォルトの動作として、そのデバイスの最初の 64k が使用されます。あるいは、`create-lu` コマンドの `meta` プロパティーを使用して、メタデータを格納するための代替ファイルを指定する方法もあります。論理ユニットのメタデータ用に十分な記憶領域 (64k を推奨) を提供できるファイルを使用することをお勧めします。

`create-tg` *group-name*
group-name という名前のターゲットグループを作成します。*group-name* は、最大長が 255 の Unicode 文字の文字列です。このグループ名は、STMF システム内で一意である必要があります。

`delete-hg` *group-name*
group-name で識別されるホストグループを削除します。

`delete-lu` *lu-name*
: `stmfadm create-lu` を使用して作成された既存の論理ユニットを削除します。これにより、事実上、その論理ユニットが STMF フレームワークからアンロードされます。その論理ユニット上の既存のデータは、すべてそのまま維持されます。

`delete-tg` *group-name*
: *group-name* で識別されるターゲットグループを削除します。

`import-lu` *lu-file*
: `stmfadm create-lu` を使用して以前に作成されたあとで `stmfadm delete-lu` を使用して STMF から削除された論理ユニットを、STMF にインポートおよびロードします。成功した場合、その論理ユニットは STMF から再度使用できるようになります。*lu-file* は、`stmfadm create-lu` コマンドで使用されるファイル名です。この論理ユニットが個別のメタデータファイルを使用している場合は、`stmfadm create-lu` コマンドで使用された `meta` プロパティー値のファイル名をここで使用する必要があります。

`list-hg` [`-v`,`--verbose`] [*host-group-name*...]
: *host-group-name* によって参照されたシステム内のホストグループの情報を一覧表示します。*host-group-name* が指定されていない場合は、システム内のすべてのホストグループが一覧表示されます。`--v` または `--verbose` オプションが指定されている場合は、ホストグループ内のすべてのメンバーが表示されます。

`list-lu` [`-v`,`--verbose`] [*lu-name*...]
: *lu-name* によって参照されたシステム内の論理ユニットの情報を一覧表示します。*lu-name* が指定されていない場合は、システム内のすべての論理ユニットが一覧表示されます。`-v` または `--verbose` オプションが指定されている場合は、論理ユニットに関する追加情報が表示されます。

`list-target` [`-v`,`--verbose`] [*target-name*...]
: *target-name* によって参照されたシステム内のターゲットポートの情報を一覧表示します。ターゲット名が指定されていない場合は、システム内のすべてのターゲットポートが一覧表示されます。`-v` または `--verbose` オプションが指定されている場合は、ログインしているイニシエータの SCSI セッション情報とともに、ターゲットに関する追加情報が表示されます。

`list-tg` [`-v`,`--verbose`] [*target-group-name*...]
: *target-group-name* によって参照されたシステム内のターゲットグループの情報を一覧表示します。*target-group-name* が指定されていない場合は、システム内のすべてのターゲットグループが一覧表示されます。`--v` または `--verbose` オプションが指定されている場合は、ターゲットグループ内のすべてのメンバーが表示されます。

`list-view` `--l`, `--lu-name` *lu-name* [*entry-name*...]
: *lu-name* によって参照された論理ユニットのビューエントリを一覧表示します。*entry-name* が指定されていない場合は、指定された論理ユニットのすべてのビューエントリが一覧表示されます。

`modify-lu` [`-p`, `--lu-prop` *logical-unit-property=val* `--s`, `--size` *size*, `-f`, `--file`] *lu-name*|*lu-file*

`stmfadm create-lu` コマンドを使用して作成された論理ユニットの属性を変更します。`-p` オプションでは、*logical-unit-property* に次のいずれかのプロパティーを指定できます。

`alias`
: 最大 255 文字。デバイスのユーザー定義の名前を表します。デフォルトはバッキングストアの名前です。

`mgmt-url`
: 最大 1024 文字。管理ネットワークアドレスの URL を表します。一対の単一引用符 (") で囲まれたスペース区切りの URL を使用すると、複数の URL を 1 つのパラメータとして渡すことができます。

`wcd`
: ライトバックキャッシュ無効。ライトバックキャッシュ無効の動作を決定するには、`true` または `false` を指定します。デフォルトは、*lu-file* 引数で指定されたバッキングストアデバイスのライトバックキャッシュ設定です。

`wp`
: 書き込み保護ビット。書き込み保護されているデバイスとして報告するかどうかを決定するには、`true` または `false` を指定します。デフォルトは `false` です。

`-s` オプションでは、*size* として、整数のあとにサイズの単位を示す次のいずれかの文字を指定します。

`k` K バイト

`m` M バイト

`g` G バイト

`t` T バイト

`p` P バイト

`e` E バイト

lu-name は、変更される論理ユニットの `guid` です。`-f` オプションが指定されている場合、オペランドはファイル名として解釈されます。これにより、現在 STMF にインポートされていない論理ユニットを変更することが可能になります。

`online-lu` *lu-name*
: 現在 STMF に登録されている論理ユニットをオンラインにします。

`online-target` *target-name*
: 指定されたターゲットをオンラインにします。

`offline-lu` *lu-name*
現在 STMF に登録されている論理ユニットをオフラインにします。

`offline-target` *target-name*
指定されたターゲットをオフラインにします。

`list-state`
STMF の動作状態および構成状態を一覧表示します。

`remove-hg-member -g` *group-name group member*
ホストグループメンバーをホストグループから削除します。*group-name* は、`create-hg` サブコマンドを使用して作成された既存のグループである必要があります。*group member* は、*name_type.name_value* として指定できます。ここで、*name_type* には次のいずれかを指定できます。

```
wwn
iqn
eui
```

また、*name_value* はイニシエータ名の値です。たとえば、ファイバチャネルイニシエータポートのワールドワイドネーム `200000e08b909221` をホストグループ `HostA` から削除するには、次のコマンドを入力します。

```
# stmfadm remove-hg-member -g HostA wwn.200000e08b909221
```

`iqn.1986-03.com.sun:01.46f7e262` という名前の ISCSI イニシエータノードメンバーを `HostA` から削除するには、次のコマンドを入力します。

```
# stmfadm remove-hg-member -g HostA iqn.1986-03.com.sun:01.46f7e262
```

あるいは、メンバーをそれぞれの SCSI 名文字列識別子で指定することもできます。上の 2 つのイニシエータをそれぞれの SCSI 名文字列識別子を使用して削除するには、次のコマンドを入力します。

```
# stmfadm remove-hg-member -g HostA eui.200000e08b909221
# stmfadm remove-hg-member -g HostA iqn.1986-03.com.sun:01.46f7e262
```

`remove-tg-member -g` *group-name group member*
ターゲットグループメンバーをターゲットグループから削除します。*group-name* は、`create-tg` サブコマンドを使用して作成された既存のグループである必要があります。*group member* は、*name_type.name_value* として指定できます。ここで、*name_type* には次のいずれかを指定できます。

```
wwn
iqn
eui
```

また、*name_value* はターゲット名の値です。たとえば、ファイバチャネルターゲットポートのワールドワイドネーム `501000e092376af7` をターゲットグループ `TG0` から削除するには、次のコマンドを入力します。

```
# stmfadm remove-tg-member -g TG0 wwn.501000e092376af7
```

iqn.1986-03.com.sun:target.01.01110 という名前の ISCSI ターゲットメンバーを TG0 から削除するには、次のコマンドを入力します。

```
# stmfadm remove-tg-member -g TG0 iqn.1986-03.com.sun:target.01.01110
```

あるいは、メンバーをそれぞれの SCSI 名文字列識別子で指定することもできます。上の 2 つのターゲットをそれぞれの SCSI 名文字列識別子を使用して削除するには、次のコマンドを入力します。

```
# stmfadm remove-tg-member -g TG0 eui.501000e092376af7
# stmfadm remove-tg-member -g TG0 iqn.1986-03.com.sun:target.01.01110
```

remove-view --l, --lu-name *lu-name entry-name*
論理ユニットの 1 つ以上のビューエントリを論理ユニットから削除します。

使用例

例 1 2 つのイニシエータポートを含むホストグループの作成

次のコマンドは、create-hg サブコマンドと add-hg-member サブコマンドを使用してホストグループを作成し、2 つのイニシエータポートをそのホストグループに追加します。

```
# stmfadm create-hg host-group-a
# stmfadm add-hg-member -g host-group-a wwn.210105b0000d92d0
```

例 2 論理ユニットへのビューエントリの追加

次のコマンドは、add-view サブコマンドを使用して、host-group-a から論理ユニットへのアクセスを許可します。

```
# stmfadm add-view -h host-group-a 6000AE40C5000000000046FC4FEA001C
```

例 3 ビューエントリの一覧表示

次のコマンドは、list-view サブコマンドを使用して、指定された論理ユニットのすべてのビューエントリを一覧表示します。

```
# stmfadm list-view -l 6000AE40C5000000000046FC4FEA001C
View Entry: 0
                Host group      : host-group-a
                Target group   : All
                LUN               : 0
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/scsi-target-mode-framework
インタフェースの安定性	確実

関連項目 hostid(1), sbdadm(1M), attributes(5)

名前

stmsboot – administration program for the Solaris I/O multipathing feature

形式

```
/usr/sbin/stmsboot [[-D (fp | mpt | mpt_sas | iscsi) ] -d | -e | -u]
      | -L | -l controller_number]
```

機能説明

The Solaris I/O multipathing feature is a multipathing solution for storage devices that is part of the Solaris operating environment. This feature was formerly known as Sun StorEdge Traffic Manager (STMS) or MPxIO.

The `stmsboot` program is an administrative command to manage enumeration of multipath–capable devices with Solaris I/O multipathing. Solaris I/O multipathing-enabled devices are enumerated under `scsi_vhci(7D)`, providing multipathing capabilities. Solaris I/O multipathing-disabled devices are enumerated under the physical controller.

In the `/dev` and `/devices` trees, Solaris I/O multipathing-enabled devices receive new names that indicate that they are under Solaris I/O multipathing control. This means a device will have a different name from its original name (after enabling) when it is under Solaris I/O multipathing control. The `stmsboot` command automatically updates `/etc/vfstab` and dump configuration to reflect the device names changes when enabling or disabling Solaris I/O multipathing. One reboot is required for changes to take effect.

オプション

The following options are supported:

`-e [ -E fp | mpt | mpt_sas | iscsi ]`
: Enables Solaris I/O multipathing on all supported multipath-capable controller ports, including `fp(7d)`, `mpt(7D)`, `mpt_sas(7D)`, and `iscsi(7D)` port drivers. Multipath-capable ports include fibre channel (`fp(7d)`) controller ports and SAS (`mpt(7D)` or `mpt_sas(7D)`) controller ports. Following this enabling, you are prompted to reboot. During the reboot, `vfstab` and the dump configuration will be updated to reflect the device name changes. Specifying `-D mpt`, `-D mpt_sas`, or `-D fp` limits the enabling operation to ports attached using the specified driver.

`-d [ -D fp | mpt | mpt_sas | iscsi ]`
: Disables Solaris I/O multipathing on all supported multipath-capable controller ports, including `fp(7d)`, `mpt(7D)`, `mpt_sas(7D)`, and `iscsi(7D)` port drivers. Multipath-capable ports include fibre channel (`fp(7d)`) controller ports and SAS (`mpt(7D)` or `mpt_sas(7D)`) controller ports. Following this disabling, you are prompted to reboot. During the reboot, `vfstab` and the dump configuration will be updated to reflect the device name changes. Specifying `-D mpt`, `-D mpt_sas`, or `-D fp` limits the disabling operation to ports attached using the specified driver.

`-u [ -D fp | mpt | mpt_sas | iscsi ]`
: Updates `vfstab` and the dump configuration after you have manually modified the configuration to have Solaris I/O multipathing enabled or disabled on specific `fp(7d)`, `mpt(7D)`, `mpt_sas(7D)`, and `iscsi(7D)` controller ports. This option prompts you to reboot. During the reboot, `vfstab` and the dump configuration will be updated to reflect the device name changes.

`-L`
: Display the device name changes from non-Solaris I/O multipathing device names to Solaris I/O multipathing device names for multipath-enabled controller ports. If Solaris I/O multipathing is not enabled, then no mappings are displayed.

`-l` *controller_number*
: Display the device name changes from non-Solaris I/O multipathing device names to Solaris I/O multipathing device names for the specified controller. If Solaris I/O multipathing is not enabled, then no mappings are displayed.

Note that `mpt_sas(7D)` has MPxIO turned on by default. This means that when using the `-L` or `-l` option with `-D mpt_sas`, `stmsboot` does not display any non-multipathed and multipathed device names.

使用法

The primary function of `stmsboot` is to control the enabling and disabling of Solaris I/O multipathing on the host. The utility automatically updates `vfstab(4)` and `dumpadm(1M)` configuration to reflect device name changes. The system administrator is responsible for modifying application configuration (for example, backup software, DBMS, and so forth) to reflect updated device names.

The `-L` and `-l` options display the mapping between multipathed and non-multipathed device names. These options function only after changes to the Solaris I/O multipathing configuration have taken effect, that is, following the reboot after invoking `stmsboot -e`.

ZFS datasets, including ZFS root datasets, are correctly handled by `stmsboot`.

使用例

例 1 Enabling Solaris I/O Multipathing

To enable Solaris I/O multipathing for all multipath-capable controllers, run:

```
# stmsboot -e
```

To enable Solaris I/O multipathing on multipath-capable `mpt(7D)` controller ports, enter:

```
# stmsboot -D mpt -e
```

To enable Solaris I/O multipathing on multipath-capable `mpt_sas(7D)` controller ports, enter:

```
# stmsboot -D mpt_sas -e
```

To enable Solaris I/O Multipathing on multipath-capable fibre channel controller ports, enter:

```
# stmsboot -D fp -e
```

To enable Solaris I/O Multipathing on multipath-capable iSCSI controller ports, enter:

```
# stmsboot -D iscsi -e
```

例 2 Disabling Solaris I/O Multipathing

To disable Solaris I/O multipathing on all multipath-capable controllers, enter:

```
# stmsboot -d
```

To disable Solaris I/O multipathing on multipath-capable mpt(7D) controller ports, enter:

```
# stmsboot -D mpt -d
```

To disable Solaris I/O multipathing on multipath-capable mpt_sas(7D) controller ports, enter:

```
# stmsboot -D mpt_sas -d
```

To disable Solaris I/O multipathing on multipath-capable iSCSI controller ports, enter:

```
# stmsboot -D iscsi -d
```

To disable Solaris I/O multipathing on multipath-capable fibre channel controller ports, enter:

```
# stmsboot -D fp -d
```

例 3 Enabling Solaris I/O Multipathing on Selected Ports

To enable Solaris I/O multipathing on specific fibre channel controller ports and disable the feature on others, manually edit the /kernel/drv/fp.conf file. (Seefp(7d).) The following command will update vfstab(4) and dumpadm(1M) configurations to reflect the changed device names:

```
# stmsboot -u
```

A similar procedure involving the /kernel/drv/mpt.conf file should be followed for devices attached by means of the mpt(7D) driver. For devices attached by means of the iscsi(7D) driver, follow a similar procedure that uses the /kernel/drv/iscsi.conf file.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os, system/library
Interface Stability	Obsolete

関連項目

dumpadm(1M), fsck(1M), mpathadm(1M), ufsdump(1M), zfs(1M), zpool(1M), ufsdump(4), vfstab(4), emlxs(7D), fcp(7D), fp(7d), iscsi(7D), mpt(7D), mpt_sas(7D), qlc(7D), scsi_vhci(7D)

Solaris SAN Configuration and Multipathing Guide (see http://docs.oracle.com)

Consult a particular storage product's system administrator's guide and release notes for further information specific to that product.

注意事項

Solaris I/O multipathing is not supported on all devices. After enabling Solaris I/O multipathing, only supported devices are placed under Solaris I/O multipathing control. Non-supported devices remain unchanged.

For Solaris releases prior to the current release, the `-e` and `-d` options replace `mpxio-disable` property entries with a global `mpxio-disable` entry in `fp.conf`.

Enabling Solaris I/O Multipathing on a Sun StorEdge Disk Array

The following applies to Sun StoreEdge T3, 3910, 3960, 6120, and 6320 storage subsystems.

To place your Sun StorEdge disk subsystem under Solaris I/O multipathing control, in addition to enabling Solaris I/O multipathing, the `mp_support` of the subsystem must be set to `mpxio` mode. The preferred sequence is to change the subsystem's `mp_support` to `mpxio` mode, then run `stmsboot -e`. If Solaris I/O multipathing is already enabled but the subsystem's `mp_support` is not in `mpxio` mode, then change the `mp_support` to `mpxio` mode and run `stmsboot -u`.

Refer to the Sun StorEdge Administrator's Guide for your subsystem for more details.

Using ufsdump

The `ufsdump(1M)` command records details of filesystem dumps in `/etc/dumpdates` (see `ufsdump(4)`). Among other items, the entries contain device names. An effect of the “active” `stmsboot` options (`-e`, `-d`, and `-u`) is to change the device name of a storage device.

Because `stmsboot` does not modify `dumpdates`, entries will refer to obsolete device names, that is, device names that were in effect before Solaris I/O multipathing configuration changes were performed. In this situation `ufsdump` will behave as if no previous dump of the filesystem had been performed. A level 0 dump will be performed.

Procedure to Use stmsboot in Conjunction with Sun Cluster

If possible, invoke `stmsboot -e` before installing Sun Cluster software. After executing `stmsboot`, install Sun Cluster software normally.

If Sun Cluster software is installed before executing `stmsboot`, follow this procedure:

On each machine in the cluster where Solaris I/O multipathing is required, execute:

```
# stmsboot -e
```

...and allow the system to reboot.

When the system comes up, enter the following two commands:

1. # **/usr/cluster/bin/scdidadm -C**
2. # **/usr/cluster/bin/scdidadm -r**

   The preceding commands update `did` mappings with new device names while preserving `did` instance numbers for disks that are connected to multiple cluster nodes. `did` instance numbers of the local disks might not be preserved. For this reason, the `did` disk names for local disks might change.

   The remaining steps are required only if local `did` number changes require editing of `/etc/vfstab`. If such editing is not required, run `/usr/cluster/bin/scgdevs` from each node in the cluster to complete the procedure.

3. Update /etc/vfstab to reflect any new did disk names for your local disks.
4. Reboot the system.

To disable the Solaris multipathing feature, use stmsboot -d (instead of stmsboot -e), then follow the procedure above.

To view mappings between the old and new device names, run stmsboot -L. To view did device name mappings, run /usr/cluster/bin/scdidadm -L.

With active-passive storage arrays, it is possible that while your host is rebooting the array controller could failover the path that a particular target is using. In this scenario, fsck(1M) will fail to open the physical path listed in /etc/vfstab. The svc:/system/filesystem/local:default SMF service will transition to a maintenance state as a result. To rectify this, consult the documentation for your storage array to failback the path. The mpathadm(1M) can assist with determining the active and passive path(s).

Limitations

On x86 platforms, the current Solaris release does not support disabling Solaris I/O multipathing of boot devices attached by means of fibre channel. Solaris I/O multipathing is always enabled for supported fibre channel-attached boot devices. Disabling Solaris I/O multipathing in this situation must be performed on a per-port basis. See fp(7d).

Executing devfsadm -C removes obsolete device entries that stmsboot relies on. This will prevent correct operation of the -d option for boot devices (regardless of platform type) and the -L option.

名前

strace – print STREAMS trace messages

形式

`strace [mid sid level]...`

機能説明

`strace` without arguments writes all STREAMS event trace messages from all drivers and modules to its standard output. These messages are obtained from the STREAMS log driver (see `log(7D)`). If arguments are provided, they must be in triplets of the form *mid, sid, level*, where *mid* is a STREAMS module ID number, *sid* is a sub-ID number, and *level* is a tracing priority level. Each triplet indicates that tracing messages are to be received from the given module/driver, sub-ID (usually indicating minor device), and priority level equal to, or less than the given level. The token `all` may be used for any member to indicate no restriction for that attribute.

The format of each trace message output is:

<*seq*> `<time>` <*ticks*> <*level*> <*flags*> <*mid*> <*sid*> <*text*>

<*seq*>	trace sequence number
`<time>`	time of message in *hh:mm:ss*
<*ticks*>	time of message in machine ticks since boot
<*level*>	tracing priority level
<*flags*>	E : message is also in the error log F : indicates a fatal error N : mail was sent to the system administrator (hardcoded as root)
<*mid*>	module ID number of source
<*sid*>	sub-ID number of source
<*text*>	formatted text of the trace message

Once initiated, `strace` will continue to execute until terminated by the user.

使用例

例 1　A sample output of the strace command:

The following example outputs all trace messages from the module or driver whose module ID is `41`:

```
strace  41 all all
```

The following example outputs those trace messages from driver or module ID `41` with sub-IDs `0`, `1`, or `2`:

```
strace  41 0 1  41 1 1  41 2 0
```

Messages from sub-IDs `0` and `1` must have a tracing level less than or equal to `1`. Those from sub-ID `2` must have a tracing level of `0`.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

attributes(5), log(7D)

『STREAMS Programming Guide』

注意事項

- There is no restriction to the number of strace processes opening the STREAMS log driver at a time.
- The log-driver records the list of the triplets specified in the command invocation, and compares each potential trace message against this list to decide if it should be formatted and sent up to the strace process. Hence, long lists of triplets will have a greater impact on overall STREAMS performance. Running strace will have the most impact on the timing of the modules and drivers generating the trace messages that are sent to the strace process. If trace messages are generated faster than the strace process can handle them, some of the messages will be lost. This last case can be determined by examining the sequence numbers on the trace messages output.

名前

strclean – STREAMS error logger cleanup program

形式

`strclean [-a age] [-d logdir]`

機能説明

`strclean` is used to clean up the STREAMS error logger directory on a regular basis (for example, by using `cron`. By default, all files with names matching `error.*` in `/var/adm/streams` that have not been modified in the last three days are removed.

オプション

The following options are supported:

`-a` *age* The maximum age in days for a log file can be changed using the `-a` option.

`-d` *logdir* A directory other than `/var/adm/streams` can be specified using the `-d` option.

使用例

例 1 A sample of using the strclean command.

This example has the same result as running `strclean` with no arguments:

```
example%  strclean -d /var/adm/streams -a 3
```

ファイル

`/var/adm/streams/error.*`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`cron(1M)`, `strerr(1M)`, `attributes(5)`

『STREAMS Programming Guide』

注意事項

`strclean` is typically run from `cron` on a daily or weekly basis.

名前

strerr – STREAMS error logger daemon

形式

`strerr`

機能説明

`strerr` receives error log messages from the STREAMS-based log driver (see `log(7D)`) and appends them to a log file. The resultant error log files reside in the directory `/var/adm/streams`, and are named `error.`*mm*`-dd`, where *mm* is the month and `dd` is the day of the messages contained in each log file.

The format of an error log message is:

<*seq*> `<time>` <*ticks*> <*flags*> <*mid*> <*sid*> <*text*>

<*seq*>	error sequence number
`<time>`	time of message in hh:mm:ss
<*ticks*>	time of message in machine ticks since boot priority level
<*flags*>	`T` : the message was also sent to a tracing process `F` : indicates a fatal error `N` : send mail to the system administrator (hardcoded as root)
<*mid*>	module ID number of source
<*sid*>	sub-ID number of source
<*text*>	formatted text of the error message

Messages that appear in the error log are intended to report exceptional conditions that require the attention of the system administrator. Those messages which indicate the total failure of a STREAMS-based driver or module should have the `F` flag set. Those messages requiring the immediate attention of the administrator will have the `N` flag set, which causes the error logger to send the message to the system administrator using `mail`. The priority level usually has no meaning in the error log but will have meaning if the message is also sent to a tracer process.

Once initiated, `strerr` continues to execute until terminated by the user. It is commonly executed asynchronously.

ファイル

`/var/adm/streams/error.`*mm*`-dd` error log file.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`attributes(5)`, `log(7D)`

『STREAMS Programming Guide』

注意事項

There is no restriction to the number of `strerr` processes opening the STREAMS–based log driver at a time.

If a module or driver is generating a large number of error messages, running the error logger will cause a degradation in STREAMS performance. If a large burst of messages are generated in a short time, the log driver may not be able to deliver some of the messages. This situation is indicated by gaps in the sequence numbering of the messages in the log files.

名前

sttydefs – maintain line settings and hunt sequences for TTY ports

形式

```
/usr/sbin/sttydefs -a ttylabel [-b] [-f final-flags]
     [-i initial-flags] [-n nextlabel]

/usr/sbin/sttydefs -l [ttylabel]

/usr/sbin/sttydefs -r ttylabel
```

機能説明

`sttydefs` is an administrative command that maintains the line settings and hunt sequences for the system's TTY ports by making entries in, and deleting entries from the `/etc/ttydefs` file.

`sttydefs` with a `-a` or `-r` option may be invoked only by the super-user. `sttydefs` with `-l` may be invoked by any user on the system.

オプション

The following options are supported:

`-a` *ttylabel*	Add a record to the `ttydefs` file, using *ttylabel* as its label. The following describes the effect of the `-b`, `-n`, `-i`, or `-f` options when used in conjunction with the `-a` option:
`-b`	Enable autobaud. Autobaud allows the system to set the line speed of a given TTY port to the line speed of the device connected to the port without the user's intervention.
`-f` *final-flags*	Specify the value to be used in the *final-flags* field in `/etc/ttydefs`. *final-flags* must be in a format recognized by the `stty` command. *final-flags* are the `termio(7I)` settings used by `ttymon` after receiving a successful connection request and immediately before invoking the service on the port. If this option is not specified, `sttydefs` will set *final-flags* equal to the `termio(7I)` flags `9600` and `sane`.
`-i` *initial-flags*	Specify the value to be used in the *initial-flags* field in `/etc/ttydefs`. *initial-flags* must be in a format recognized by the `stty` command. These flags are used by `ttymon` when searching for the correct baud rate. They are set prior to writing the prompt. If this option is not specified, `sttydefs` will set *initial-flags* equal to the `termio(7I)` flag `9600`.
`-n` *nextlabel*	Specify the value to be used in the *nextlabel* field in `/etc/ttydefs`. If this option is not specified, `sttydefs` will set *nextlabel* equal to *ttylabel*.
`-l`[*ttylabel*]	If a *ttylabel* is specified, `sttydefs` displays the record from `/etc/ttydefs` whose TTY label matches the specified *ttylabel*. If no *ttylabel* is specified, `sttydefs` displays the entire contents of `/etc/ttydefs`. `sttydefs` verifies that each entry it displays is correct and that the entry's *nextlabel* field references an existing
`-r` *ttylabel*	Remove any record in the `ttydefs` file that has *ttylabel* as its label.

Output

If successful, `sttydefs` will exit with a status of `0`. `sttydefs -l` will generate the requested information and send it to standard output.

使用例

例 1　A sample of sttydefs command.

The following command lists all the entries in the `ttydefs` file and prints an error message for each invalid entry that is detected.

```
example# sttydefs -l
```

The following shows a command that requests information for a single label and its output:

```
example# sttydefs -l 9600
-----------------------------------------------------------------
9600:9600 hupcl erase ^h:9600 sane ixany tab3 hupcl erase ^h::4800
-----------------------------------------------------------------
ttylabel:    9600
initial flags:    9600 hupcl erase ^h
final flags:    9600 sane ixany tab3 hupcl erase ^h
autobaud:    no
nextlabel:      4800
```

The following sequence of commands will add the labels `1200`, `2400`, `4800`, and `9600` and put them in a circular list:

```
sttydefs -a 1200 -n 2400 -i 1200 -f "1200 sane"
sttydefs -a 2400 -n 4800 -i 2400 -f "2400 sane"
sttydefs -a 4800 -n 9600 -i 4800 -f "4800 sane"
sttydefs -a 9600 -n 1200 -i 9600 -f "9600 sane"
```

ファイル

`/etc/ttydefs`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`attributes(5)`, `termio(7I)`

名前

su – スーパーユーザーまたは別のユーザーに変更

形式

su [-] [*username* [arg...]]

機能説明

su コマンドを使用すると、ログオフをしないで別のユーザーまたは別の役割になることができます。デフォルトの *username* は root (スーパーユーザー) です。

su を使用するときは、適切なパスワードを入力する必要があります (ユーザーがすでに root の場合は不要)。パスワードが正しければ、指定した *username* 用に設定されている実ユーザー ID、実効ユーザー ID、グループ ID、補助グループリストを持つ新しいシェルプロセスが生成されます。また、新しいシェルのプロジェクト ID は、指定されたユーザーのデフォルトのプロジェクト ID に設定されます。getprojent(3PROJECT) および setproject(3PROJECT) を参照してください。新しいシェルは、*username* のパスワードファイルエントリ (passwd(4) を参照) のシェルフィールドで指定されたシェルです。シェルが指定されていないと、/usr/bin/sh (sh(1) を参照) になります。スーパーユーザー特権が必要な場合に、exec(2)によってスーパーユーザーのシェルを起動できないときは、代わりに /sbin/sh が使用されます。元のユーザー ID の特権に戻る場合は、EOF 文字 (CTRL-D) を入力して新しいシェルを終了します。

コマンド行に引数を指定すると、その引数が新しいシェルに渡されます。sh などのプログラムを実行している場合は、*arg* (引数) に -c *string* と指定すると、シェルによって *string* が実行されます。また、引数に -r を指定すると、ユーザーに制限付きのシェルが与えられ提供されます。

ログイン環境を作成するために、コマンド「"su –"」は次の作業を実行します。

- すでに伝播されている環境変数に加えて、指定されたユーザーの環境から環境変数 LC* と LANG を伝播します。
- ユーザーの環境から環境変数 TZ を伝播します。環境変数 TZ がユーザーの環境に存在しない場合、su は、/etc/default/login にある TIMEZONE パラメータの TZ 値を使用します。
- MAIL を /var/mail/*new_user* に設定します。

su の最初の引数に – (ダッシュ) を指定すると、指定したユーザーとして実際にログインした場合と同じ環境が渡されます。そうしない場合、$PATH 以外の環境が渡されます。$PATH は /etc/default/su 中で PATH と SUPATH によって制御されます。

su を実行して別のユーザーになる操作は、すべてログファイル /var/adm/sulog に記録されます (sulog(4) を参照)。

セキュリティー

su は、サービス名 su で pam(3PAM) を使って、認証、アカウント管理、および資格の確立を行います。

使用例

例 1　以前にエクスポートした環境を維持してユーザー bin になる

以前にエクスポートした環境のままユーザー bin になるには、以下のコマンドを実行します。

```
example% su bin
```

例 2　ユーザー bin になり bin のログイン環境に変更する

ユーザー bin になり、bin がログインした場合と同じ環境に切り替えるには、次のコマンドを実行します。

```
example% su - bin
```

例 3　ユーザー bin の環境とアクセス権でコマンドを実行する

ユーザー bin の一時的な環境とアクセス権を使ってコマンド command を実行する場合は、次のように入力します。

```
example% su - bin -c "command args"
```

環境変数

セキュリティー上の理由から、接頭辞 LD_ が付いている環境変数は削除されました。このため、以前にユーザー bin でエクスポートした接頭辞 LD_ 付きの環境変数は、su bin を実行しても使用できません。

LC_* 変数 (LC_CTYPE、LC_MESSAGES、LC_TIME、LC_COLLATE、LC_NUMERIC、および LC_MONETARY) (environ(5) 参照) のいずれも環境に設定されていなければ、それぞれ対応するロケールのカテゴリにおける su の動作は、環境変数 LANG によって決定されます。LC_ALL が設定されていれば、その内容が LANG 変数やその他の LC_* 変数より優先されます。上記の変数がどれも設定されていなければ、「C」ロケール (米国の形式) によって su の動作が決定します。

LC_CTYPE　　su が文字を処理する方法を決定します。LC_CTYPE に有効な値が設定されていると、su は、そのロケールにあった文字を含むテキストやファイル名を表示および処理できます。また、拡張 UNIX コード (EUC) の表示または処理も可能です (1 ～ 3 バイト幅の文字を使用)。また、su は 1、2、またはそれ以上のカラム幅の EUC 文字も処理することができます。「C」ロケールにおいては、ISO 8859-1 の文字だけが有効です。

LC_MESSAGES　　診断メッセージや情報メッセージの表示方法を決定します。また、メッセージの言語とスタイル、肯定・否定の応答形式も決定します。「C」ロケールでは、メッセージはプログラム自身が使用しているデフォルトの形 (通常、米語) で表示されます。

ファイル

$HOME/.profile
　sh および ksh ユーザーのログインコマンド

/etc/passwd
　システムのパスワードファイル

/etc/profile
: システム全体の sh と ksh のログインコマンド

/var/adm/sulog
: ログファイル

/etc/default/su
: このファイル中に指定できるデフォルトパラメータは以下のとおりです。

SULOG
: 定義されている場合、別のユーザーになるための su の試行はすべて、指定されたファイルに記録されます。

CONSOLE
: 定義されている場合、root になるための su の試行はすべて、コンソールに記録されます。

PATH
: デフォルトパス(/usr/bin:)

SUPATH
: root になるために su を起動しているユーザーのデフォルトパス(/usr/sbin:/usr/bin)

SYSLOG
: すべての su 試行を記録するのに syslog(3C) の LOG_AUTH を使うかどうかを指定します。LOG_NOTICE メッセージは root への su に対して生成され、LOG_INFO メッセージはその他のユーザーへの su に対して生成されます。LOG_CRIT メッセージは、su に失敗した場合に生成されます。

/etc/default/login
: このファイル中に指定できるデフォルトパラメータは以下のとおりです。

SLEEPTIME
: 存在する場合、ログイン失敗が画面に出力されて、別のログイン試行が許可されるまで待機する秒数を設定します。デフォルトは 4 秒。最小は 0 秒。最大は 5 秒。

  su と login(1) は両方とも SLEEPTIME の値の影響を受けます。

TIMEZONE
: シェルの TZ 環境変数を設定します。

属性

次の属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

csh(1), env(1), ksh(1), login(1), roles(1), sh(1), syslogd(1M), exec(2), getprojent(3PROJECT), setproject(3PROJECT), pam(3PAM), pam_authenticate(3PAM), pam_acct_mgmt(3PAM), pam_setcred(3PAM), pam.conf(4), passwd(4), profile(4), sulog(4), syslog(3C), attributes(5), environ(5)

名前

sulogin – access single-user mode

形式

`sulogin`

機能説明

The `sulogin` utility is automatically invoked by `init` when the system is first started. It prompts the user to type a user name and password to enter system maintenance mode (single-user mode) or to type EOF (typically CTRL-D) for normal startup (multi-user mode). The user should never directly invoke `sulogin`. The user must have the `solaris.system.maintenance` authorization.

The `sulogin` utility can prompt the user to enter the root password on a variable number of serial console devices, in addition to the traditional console device. See `consadm(1m)` and `msglog(7D)` for a description of how to configure a serial device to display the single-user login prompt.

ファイル

`/etc/default/sulogin`

Default value can be set for the following flag:

PASSREQ

Determines if login requires a password. Default is `PASSREQ=YES`.

`/etc/default/login`

Default value can be set for the following flag:

`SLEEPTIME`

If present, sets the number of seconds to wait before login failure is printed to the screen and another login attempt is allowed. Default is `4` seconds. Minimum is `0` seconds. Maximum is `5` seconds.

Both `su(1M)` and `login(1)` are affected by the value of `SLEEPTIME`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`auths(1)`, `login(1)`, `consadm(1m)`, `init(1M)`, `su(1M)`, `attributes(5)`, `msglog(7D)`

注意事項

By default, the root user has all authorizations.

Granting the `solaris.system.maintenance` authorization to the Console User Rights Profile may have an undesirable side effect of granting the currently logged in user maintenance mode access. The `solaris.system.maintenance` authorization should be directly granted to appropriate users rather than through the Console User Rights Profile.

名前

svadm – Availability Suite ストレージボリュームの操作を制御するためのコマンド行インタフェース

形式

```
svadm -h

svadm -v

svadm [-C tag]

svadm [-C tag] -i

svadm [-C tag] -e {-f config_file | volume}

svadm [-C tag] -d {-f config_file | volume}

svadm [-C tag] -r {-f config_file | volume}
```

機能説明

svadm コマンドは、指定されたボリュームのストレージボリューム (SV) ドライバを有効または無効にしたり、システムを動的に再構成したりする機能を提供することによって SV ドライバを制御します。

オプション

svadm コマンドの引数を指定しない場合、このユーティリティーは、現在 SV の制御下にあるボリュームの一覧を表示します。svadm は、次のオプションをサポートします。

-C *tag*
: クラスタ化されたノードでは、tag で指定されたクラスタリソースグループまたはディスクグループ名に属するボリュームのみに操作を制限します。このオプションは、クラスタ化されていないシステム上では使用できません。特殊な local を使用すると、クラスタ内のほかのノードに切り替えることのできないボリュームのみに操作を制限できます。

-d
: コマンド行または構成ファイルで指定された SV デバイスを無効にします。このオプションで -C タグが指定されている場合は、ボリュームがこのクラスタディスクグループ内に存在する必要があります。

-e
: コマンド行または構成ファイルで指定された SV デバイスを有効にします。このボリュームの詳細が現在の構成内に保存されます。dscfg(1M) を参照してください。このオプションで -C タグが指定されている場合は、ボリュームがこのクラスタディスクグループ内に存在する必要があります。

-f *config_file*
: ボリュームの一覧を含む構成ファイルを指定します。コマンドは、このボリュームの一覧を読み取ってから操作を実行します。*config_file* は、ボリュームのパス名を 1 行に 1 つずつ記載した単純なリスト形式になります。空白の行、およびコメント文字 (#) で始まる行は無視されます。

-h
: svadm の使用法の要約を表示します。

-i
現在 SV の制御下にあるボリュームの詳細なステータスを表示します。

-r
config_file が指定されている場合は、実行中のシステムを *config_file* で指定された構成に一致するように再構成します。-c オプションが指定されている場合は、各ボリュームのクラスタタグを比較し、それを *cluster_tag* に変更します。このオプションでボリュームが指定されている場合、このオプションは、そのボリュームに関連付けられたクラスタタグを再構成する場合にのみ有効になります。-e または -d オプションは、1 つのボリュームを有効または無効にする場合に使用するようにしてください。

-v
SV のバージョン番号を表示します。

使用法

SV ボリュームが有効になっている場合、デバイスへの通常のシステムコールアクセス (Intro(2) を参照) は StoreEdge アーキテクチャーソフトウェアにリダイレクトされます。これにより、標準アプリケーションで Sun StorageTek ポイントインタイムコピーや遠隔ミラーソフトウェアなどの StorageTek 機能を使用できるようになります。

svadm コマンドは、有効化 (-e) 操作と無効化 (-d) 操作を実行するときに、Availability Suite ログファイル /var/adm/ds.log (ds.log(4) を参照) 内にエントリを生成します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	driver/storage/sv、driver/storage/sv
インタフェースの安定性	確実

関連項目

dscfg(1M), ds.log(4), attributes(5), sv(7D)

名前

svcadm – サービスインスタンスの操作

形式

```
/usr/sbin/svcadm [-v] enable [-rst] {FMRI | pattern}...

/usr/sbin/svcadm [-v] disable [-st] {FMRI | pattern}...

/usr/sbin/svcadm [-v] restart {FMRI | pattern}...

/usr/sbin/svcadm [-v] refresh {FMRI | pattern}...

/usr/sbin/svcadm [-v] clear {FMRI | pattern}...

/usr/sbin/svcadm [-v] mark [-It] instance_state
     {FMRI | pattern}...

/usr/sbin/svcadm [-v] milestone [-d] milestone_FMRI
```

機能説明

svcadm は、サービス管理機能 (smf(5) を参照) 内で実行中のサービスに対するアクションの要求を発行します。サービスに対するアクションは、それに割り当てられているサービスリスタータエージェントによって実行されます。デフォルトのサービスリスタータは svc.startd です (svc.startd(1M) を参照)。

オプション

次のオプションがサポートされています。

-v　アクションを詳細形式で標準出力に出力します。

サブコマンド

一般的な操作

次に示すサブコマンドは、サービスインスタンスの一般的な管理に使用されます。

1 つ以上のオペランドを取るサブコマンドの場合、オペランドがサービスインスタンスではなくサービスを指定しており、かつそのサービスのインスタンスが 1 つだけ存在するときは、svcadm はそのインスタンスを操作します。省略形の *FMRI* (障害管理リソース識別子) またはパターンが複数のサービスに一致する場合は、警告メッセージが表示され、そのオペランドは無視されます。smf(5) を参照してください。

サービスに複数のインスタンスが存在する場合、svcadm はゼロ以外の終了ステータスを返します。

enable [-rst] {*FMRI* | *pattern*}...　オペランドで指定されているサービスインスタンスを有効にします。割り当てられているリスタータは、各サービスインスタンスをオンライン状態にしようとします。このアクションには、サービスインスタンスの「general」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。

-r オプションが指定されている場合、svcadm は各サービスインスタンスを有効にし、その依存関係を再帰的に有効にします。

-s オプションが指定されている場合、svcadm は各サービスインスタンスを有効にしたあと、各サービスインスタンスが online または degraded 状態になるまで待ちます。管理者の操作なしではサービスがこれらの状態に到達できないと判定した場合、svcadm は状態を待たずに戻ります。

-t オプションが指定されている場合、svcadm は各サービスインスタンスを一時的に有効にします。一時的な有効はリブートまでに限り継続します。このアクションには、サービスインスタンスの「restarter_actions」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。デフォルトでは、enable はリブート後も永続します。

disable [-st] {*FMRI* | *pattern*}...

オペランドで指定されているサービスインスタンスを無効にします。割り当てられているリスタータは、各サービスインスタンスを無効状態にしようとします。このアクションには、サービスインスタンスの「general」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。

-s オプションが指定されている場合、svcadm は各サービスインスタンスを無効にしたあと、各サービスインスタンスが無効状態になるまで待ちます。管理者の操作なしではサービスがこの状態に到達できないと判定した場合、svcadm は状態を待たずに戻ります。

-t オプションが指定されている場合、svcadm は各サービスインスタンスを一時的に無効にします。一時的な無効はリブートまでに限り継続します。このアクションには、サービスインスタンスの「restarter_actions」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。デフォルトでは、disable はリブート後も永続します。

restart {*FMRI* | *pattern*}...

オペランドで指定されているサービスインスタンスの再起動を要求します。このアクションには、サービスインスタンスの「restarter_actions」プロパティーグループを

変更するためのアクセス権が必要です (smf_security(5) を参照)。サービスの再起動は、ほとんどのリスタータではサービスの完全「停止」後の「起動」として実装されています。

このサブコマンドは、オンライン (online) 状態または機能低下 (degraded) 状態にあるサービスだけを再起動できます。それらの状態は、smf(5) に定義されています。

refresh {*FMRI* | *pattern*}...

オペランドで指定されている各サービスインスタンスについて、サービスの実行中の構成のスナップショットを現在の構成の値で更新するよう、割り当てられているリスタータに要求します。これらの値の一部はすぐに有効になります (たとえば、依存関係の変更など)。その他の値は、次回のサービスの restart まで有効になりません。詳細については、リスタータおよびサービスのドキュメントを参照してください。

サービスが svc.startd(1M) によって管理されている場合は、refresh メソッドが存在していればそれが呼び出され、サービスに自身の構成を読み込み直すように要求します。その他のリスタータについては、リスタータのドキュメントを参照してください。

このアクションには、サービスインスタンスの「restarter_actions」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。

clear {*FMRI* | *pattern*}...

オペランドで指定されている各サービスインスタンスについて、インスタンスが maintenance 状態の場合は、割り当てられているリスタータにサービスが修復されたことを通知します。インスタンスが degraded 状態の場合は、サービスを online 状態にするよう、割り当てられているリスタータに要求します。このアクションには、サービスインスタンスの「restarter_actions」プロパティーグループを変更するためのアクセス権が必要です (smf_security(5) を参照)。

例外的な操作

次に示すサブコマンドは、サービスの開発および一時的な管理操作に使用されます。

`mark [-It]` *instance_state* {*FMRI* | *pattern*}...

instance_state が「maintenance」の場合、`svcadm` は、オペランドで指定されている各サービスについて、サービスを `maintenance` 状態にするよう、割り当てられているリスタータに要求します。各リスタータに対して実行されるアクションの詳細については、`svc.startd(1M)` および `inetd(1M)` を参照してください。

instance_state が「degraded」の場合、`svcadm` は、オペランドで指定されているサービスのうちオンライン状態のものについて、サービスを `degraded` 状態にするよう、サービスに割り当てられているリスタータに要求します。

`-I` オプションが指定されている場合、要求には「即時 (immediate)」としてフラグが設定されます。

`-t` オプションは保守要求にのみ有効です。このオプションが指定されている場合、要求には「一時 (temporary)」としてフラグが設定され、その効果は次回のリブートまでに限り継続します。

`milestone [-d]` *milestone_FMRI*

milestone_FMRI がキーワード「none」の場合は、マスターリスタータ `svc:/system/svc/restarter:default` を除くすべてのサービスが一時的に無効になります。

milestone_FMRI がキーワード「all」の場合は、すべてのサービスについて一時的な enable 要求および disable 要求が無効化されます。

milestone_FMRI が次のいずれかの場合:

```
svc:/milestone/single-user:default
svc:/milestone/multi-user:default
svc:/milestone/multi-user-server:default
```

指定されたサービスおよびそれが (直接または間接的に) 依存しているすべてのサービスについて、一時的な enable 要求および disable 要求が無効化されます。その他のサービスはすべて一時的に無効になります。

システムの現在のマイルストーンを milestone サブコマンドで変更する場合、システムの現在の実行レベルは変更されません。システムの実行レベルを変更するには、`/usr/sbin/init` を直接呼び出します。

このアクションには、`svc:/system/svc/restarter:default` サービスインスタンスの「options_ovr」プロパティーグループを変更するためのアクセス権が必要です (`smf_security(5)` を参照)。

`-d` オプションが指定されている場合は、要求されたマイルストーンへの変更がすぐに、前述のとおり実行されます。また、指定されたマイルストーンはデフォルトのブートマイルストーンとなり、リブート後も持続します。デフォルトのマイルストーンは、マスターリスタータ `svc:/system/svc/restarter:default` の `options/milestone` プロパティーで定義されます。このプロパティーが存在しない場合、デフォルトは「all」です。このアクションには、`svc:/system/svc/restarter:default` サービスインスタンスの「options」プロパティーグループを変更するためのアクセス権が必要です (`smf_security(5)` を参照)。

オペランド

次のオペランドがサポートされています。

FMRI

1 つまたは複数のインスタンスを指定する *FMRI*。インスタンス名を指定するか、サービス名の連続する部分を指定することにより、*FMRI* を省略形にすることもできます。たとえば、次のような *FMRI* を仮定します。

```
svc:/network/smtp:sendmail
```

この場合、次に示す省略形はすべて有効です。

```
sendmail
:sendmail
smtp
smtp:sendmail
network/smtp
```

これに対し、次のものは無効です。

```
mail
network
network/smt
```

FMRI がサービスを指定している場合、コマンドはそのサービスのすべてのインスタンスに適用されます。*FMRI* の省略形は不安定なので、スクリプトやその他の長期に渡って使用するツールには使用しないでください。

pattern

`fnmatch(5)` で説明されている展開規則に従ってサービスインスタンスの *FMRI* と照合されるパターン。このパターンが「svc:」で始まっていない場合は、「svc: /」が先頭に付加されます。

省略形の *FMRI* またはパターンが複数のサービスに一致する場合は、警告メッセージが表示され、そのオペランドは無視されます。

使用例

例 1　サービスインスタンスの再起動

次のコマンドは NFS サーバーを再起動します。デフォルトのサービスインスタンスの完全な *FMRI* は次のとおりです。 `svc:/network/nfs/server:default`

ただし、完全な *FMRI* を次のような省略形にすることもできます。

```
# svcadm restart nfs/server
```

例 2　標準 HTTP サーバーの無効化

次のコマンドは、省略形の *FMRI* を使用して標準 HTTP サーバーを無効にします。

```
$ svcadm disable http
```

例3 インスタンスおよびその依存インスタンスの有効化

次のコマンドは、foo:bar インスタンスおよびそれが依存しているすべてのインスタンスを有効にします。

```
$ svcadm enable -r foo:bar
```

例4 インスタンスの同期的な有効化

次のコマンドは foo:bar インスタンスを有効にします。インスタンスがオンラインになるまで、あるいはサービスがオンライン状態に到達できないと svcadm が判定するまで、コマンドは戻りません。

```
$ svcadm enable -s foo:bar
```

例5 実行中のサービスの制限と復元

次のコマンドは、実行中のサービスをシングルユーザーモードに制限します。

```
# svcadm milestone milestone/single-user
```

次のコマンドは、実行中のサービスを復元します。

```
# svcadm milestone all
```

終了ステータス

次の終了値が返されます。

0 正常終了。

1 致命的なエラーが発生しました。1 つ以上のエラーメッセージが標準エラーに表示されます。

2 無効なコマンド行オプションが指定されました。

3 svcadm は、サービスインスタンス自体の問題により、待機中のサービスインスタンスが管理者の操作なしでは目的の状態に到達できないと判定しました。

4 svcadm は、サービスの依存関係の問題により、待機中のサービスインスタンスが管理者の操作なしでは目的の状態に到達できないと判定しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照。

対話型出力は「不確実」です。呼び出しおよび非対話型出力は「確実」です。

関連項目

svcprop(1), svcs(1), inetd(1M), init(1M), svccfg(1M), svc.startd(1M), libscf(3LIB), contract(4), attributes(5), smf(5), smf_security(5)

注意事項

svcadm がサービスおよびその依存関係の状態の変化を待機する時間は、それらのメソッドのタイムアウトによって暗黙に制限されます。たとえば、サービスがデフォルトのリスタータを使用している場合、その start メソッドがハングアップすると、タイムアウトの期限が切れたときサービスは保守状態に移行します。すると svcadm は、管理者の操作なしではこのサービスがオンライン状態に到達できないと判定します。

ファイルに (直接または間接的に) 依存しているサービスを同期的に有効にしようとすると、そのファイルの入っているディレクトリを検索するために必要な特権を呼び出し元が持っていない場合は、依存関係が満たされていないことを示す終了ステータスで失敗する場合があります。この制限事項は、Solaris の将来のリリースでは取り除かれる可能性があります。

名前

svccfg – サービス構成のインポート、エクスポート、および変更

形式

```
/usr/sbin/svccfg [-v] [-s FMRI]

/usr/sbin/svccfg [-v] [-s FMRI] subcommand [args]...

/usr/sbin/svccfg [-v] [-s FMRI] -f command-file
```

機能説明

svccfg コマンドは、サービス構成リポジトリ内のデータを操作します。svccfg は、個別のサブコマンドを使用するか、一連のサブコマンドが格納されているコマンドファイルを指定することで、対話形式で呼び出すことができます。

通常、リポジトリ内の既存のサービスに加えた変更は、次にサービスインスタンスが更新されるまでサービスに反映されません。詳細については、svcadm(1M) マニュアルページの refresh サブコマンドを参照してください。

オプション

次のオプションがサポートされています。

-f *command-file*
: svccfg サブコマンドを *command-file* から読み込んで実行します。

-s *FMRI*
: サブコマンドを実行する前に、*FMRI* (障害管理リソース識別子) で指定されているエンティティーを選択します。smf(5) を参照してください。

-v
: 冗長。

サブコマンド

サブコマンドは、あとに続くサブセクションに指定されたカテゴリに分けられます。

FMRI を受け入れるサブコマンドはすべて、省略形や展開パターンも受け入れます。インスタンスおよびサービスは、インスタンス名、またはサービス名の最後の部分を指定する方法で、省略して入力することができます。たとえば、次のような *FMRI* を仮定します。

```
svc:/network/smtp:sendmail
```

この場合、次に示す省略形はすべて有効です。

```
sendmail
:sendmail
smtp
smtp:sendmail
network/smtp
```

これに対し、次のものは無効です。

```
mail
network
network/smt
```

FMRI の省略形は不安定なので、スクリプトやその他の長期に渡って使用するツールには使用しないでください。パターンが複数のインスタンスまたはサービスに一致する場合は、エラーメッセージが表示され、処理は行われません。

一般的なサブコマンド

`end`
`exit`
`quit`
すぐに終了します。

`repository` [`-p` *prefix*] *repfile*
repfile をリポジトリとして使用します。デフォルトでは、`svccfg`(1M) はシステムリポジトリを使用します。

リポジトリは、現在実行しているのと同一バージョンの Solaris のファイル (パッチを含む) とのみ使用してください。このサブコマンドをシステムリポジトリ `/etc/svc/repository.db` とともに使用しないでください。

`svccfg repository` を使用して、導入時間の前に SMF リポジトリを事前に生成する場合は、`-p` オプションを使用して、`import` でインポートされたマニフェストのシステム標準の場所にルートプレフィックスを指定します。リポジトリを稼働中のシステムに配置すると、このプレフィックスは `/lib/svc/manifest` および `/var/svc/manifest` で置き換えられます。`-p` ディレクトリにあるマニフェストが実行時にシステム標準の場所に表示されない場合は、これに関連付けられたサービスが削除されます。

`set` [`-v`|`-V`]
オプション動作を設定します。オプションが何も指定されていない場合、set は現在有効になっているオプションを表示します。

`-v`
詳細表示モードを有効にします。

`-V`
詳細表示モードを無効にします。

サービスマニフェストおよびプロファイルのサブコマンド

`apply` [`-n`] [`-v`] *file* | *directory*
引数がサービスプロファイルまたはマニフェストの場合は、構成を SMF リポジトリの管理レイヤーに適用します。必要に応じて、サービス、インスタンス、プロパティーグループ、およびプロパティーが作成されます。

あるプロパティーまたはプロパティーグループのタイプ属性を指定しなかった場合、既存のタイプ設定やサービステンプレートからそのタイプを判定する試みが行われます。タイプの決定に失敗すると警告が発せられ、サービスやインスタンスに不整合なデータが導入されないようにそのサービスがスキップされます。存在しないサービスやインスタンスは無視されます。

プロファイル内で制限が緩和された要素定義を使用するには、`DOCTYPE` エントリに次の定義を追加する必要があります。

```
<!ENTITY % profile "INCLUDE">
<!ENTITY % manifest "IGNORE">
```

プロファイルによって変更されたサービスやインスタンスは更新されます。-n を指定した場合、プロファイルは処理されますが、変更は SMF リポジトリに適用されません。構文エラーが見つかると必ず標準エラー出力に報告され、終了コード 1 が返されます。サービスプロファイルについては、smf(5) を参照してください。このコマンドでは、サービスやインスタンスのプロパティーを変更する特権が必要です。プロパティーを変更するために必要な特権については、smf_security(5) を参照してください。

マニフェスト内およびリポジトリ内のテンプレートデータに基づいてマニフェストまたはプロファイル内のサービスやインスタンスが検証され、テンプレート違反が検出されると必ず警告が発行されます。テンプレートの説明については、smf_template(5) を参照してください。-V オプションを指定した場合、定義されているテンプレートに違反したマニフェストのインポートが失敗します。svccfg の対話型呼び出しでは、-V がデフォルトの動作になります。

apply の引数がディレクトリである場合、そのディレクトリツリーの下で見つかったすべてのプロファイルが前述のように適用されます。指定したファイルや、指定したディレクトリの下で見つかったいずれかのファイルがサービスプロファイルでなかった場合、このサブコマンドは失敗します。

extract [-a] [-l *layer*] [>*file*]

システム上のローカルカスタマイズを表示するサービスプロファイルを出力します。

-l を指定すると、値をエクスポートするレイヤーの一覧を選択できます。-l オプションにはレイヤー名が必要であり、引数として manifest、system-profile、site-profile、admin、current、all を指定します。current と all は同義語であり、最上位レイヤー値を意味します。listprop サブコマンドでは整合性のために両方が提供されます。 複数のレイヤーを指定するには、コンマで区切るか、複数の -l オプションを使います。

-l が指定されていない場合、デフォルトは -l admin,site-profile です。

複数選択されたレイヤーでプロパティーが定義されている場合は、最上位レイヤーのみがプロファイルでエクスポートされます。

-a オプションを指定しないと、保護された情報 (read_authorization プロパティーの存在によって識別される。smf_security(5) を参照) を含むプロパティーグループがプロパティー値なしで抽出されます。-a オプションを指定すると、すべての値が抽出されます。これらの値を読み取るための特権が不十分である場合は、エラーが発生します。

export [-a] *service_FMRI* [>*file*]

svccfg export の実行は以下と同等です。

```
svccfg extract -l current [-a] service_FMRI [>file]
```

import [-V] [*file* | *directory*]

システムで管理されたファイルシステムの場所 (/lib/svc/manifest および /var/svc/manifest のサブディレクトリ) にあるファイルに svccfg import を配置すると、svcadm restart manifest-import が起動されます。

マニフェストをインポートするには、システムで管理された場所にマニフェストを配置し、svcadm restart manifest-import を起動する方法をお勧めします。

その他の場所にあるファイルに svccfg import を配置すると、プロパティーが管理カスタマイズとして管理レイヤーにインポートされます。 以下の処理と同じ結果になります。

svccfg apply [*file* | *directory*]

inventory *file*

file がサービスマニフェストまたはプロファイルであると判定された場合は、file で指定されているサービスおよびインスタンスの *FMRI* が出力されます。各サービスについて、そのインスタンスの FMRI に続いてサービスの FMRI が表示されます。

validate [*file* | *fmri*]

validate サブコマンドで処理できるのは、マニフェストファイル、インスタンス FMRI、現在のインスタンスエンティティー選択、現在のスナップショットエンティティー選択のいずれかです。引数が指定された場合、svccfg はその指定されたファイルが存在するかどうかをチェックします。ファイルが存在する場合は、ファイルが検証されます。指定された名前のファイルが存在しない場合、その引数は FMRI パターンとして処理されます。ファイル名と FMRI との間で競合が発生する場合には、svc: および file: 接頭辞を使用して引数の解釈方法を svccfg に知らせます。

ユーザーがファイルを指定した場合、そのファイルは import -V と同様の方法で処理されますが、リポジトリの変更は一切行われません。エラーが検出された場合、svccfg はエラーを表示し、ゼロ以外の終了ステータスで終了します。

インスタンス *fmri*、インスタンスエンティティー選択、スナップショットエンティティー選択のいずれかの場合、指定された合成形式のインスタンス (smf(5) の「プロパティーおよびプロパティーグループ」を参照) が、リポジトリ内のテンプレートデータに基づいて検証されます。インスタンス FMRI とインスタンスエンティティー選択の場合、「実行中の」スナップショットが検証に使用されます。テンプレート違反が検出されると必ず警告が発行されます。テンプレートの説明については、smf_template(5) を参照してください。

サービスまたはインスタンスがマニフェストまたはプロファイル内にある場合、そのサービスまたはインスタンスは非表示になります。smf(5) を参照してください。

エンティティーの選択、変更、およびナビゲーションのサブコマンド

「エンティティー」とは、スコープ、サービス、またはサービスインスタンスを指します。

add *name*
現在の選択内容の子として、指定された名前で新しいエンティティーが作成されます。エンティティーを作成するために必要な特権については、smf_security(5) を参照してください。

delete [-f] {*name* | *fmri*}
name で指定されている現在の選択内容の子、または fmri で指定されているエンティティーが削除されます。-f フラグを指定しない限り、「online」状態または「degraded」状態のサービスインスタンスを削除しようとすると失敗します。サービスまたはサービスインスタンスが「framework」タイプの「dependents」プロパティーグループを持っている場合は、その中の「astring」タイプまたは「fmri」タイプの各プロパティーが調べられ、そのプロパティーがサービスまたはサービスインスタンスの名前を指定する 1 つの値を持っているときは、指定されたサービスまたはサービスインスタンスの、プロパティーと同じ名前を持つ dependency プロパティーグループが削除されます。サービス構成を削除するために必要な特権については、smf_security(5) を参照してください。

list [*pattern*]
現在の選択内容の子エンティティーのうち、 glob パターン *pattern* に一致する名前を持つものが表示されます (fnmatch(5) を参照)。 プロパティーを持つエンティティー、すなわちサービスおよびサービスインスタンスには、「:properties」も一覧表示されます。

select {*name* | *fmri*}
引数で現在の選択内容の子が指定されている場合は、それが現在の選択内容になります。それ以外の場合、引数は FMRI であると解釈され、引数で指定されているエンティティーが現在の選択内容になります。

unselect
現在の選択内容の親が現在の選択内容になります。

プロパティーの検査および変更のサブコマンド

addpg *name type* [*flags*]
指定された名前 *name* と type で、プロパティーグループを現在の選択内容に追加します。*flags* は、プロパティーグループの作成に使用するフラグを指定する文字列です。「P」は SCF_PG_FLAG_NONPERSISTENT を表します (scf_service_add_pg(3SCF) を参照)。プロパティーグループを作成するために必要な特権については、smf_security(5) を参照してください。

addpropvalue *pg/name* [*type:*] *value*
指定された値をプロパティーに追加します。*type* が指定されている場合で、プロパティーが存在しているときは、そのプロパティーのタイプが *type* と一致しないと、このサブコマンドは失敗します。値は二重引用符で囲まれる場合があります。二重引用符やバックスラッシュを含んでいる文字列値は、二重引用符で囲む必要があります。また、文字列値に含まれている二重引用符とバックス

ラッシュは、バックスラッシュでエスケープする必要があります。存在していないプロパティーは作成されますが、その場合は *type* 指定子が指定されている必要があります。使用可能なプロパティータイプのリストについては、scf_value_create(3SCF) を参照してください。プロパティーを変更するために必要な特権については、smf_security(5) を参照してください。新しい値は、プロパティーに関連付けられたプロパティー値のリストの末尾に追加されます。

delcust [-M] [*pattern*]
: 現在選択されている管理カスタマイズを削除します。引数が指定されている場合、それは glob パターンと見なされ、引数に一致する名前を持つプロパティーグループおよびプロパティーだけが削除されます。

  -M が指定されている場合は、「マスクされた」エンティティーのみが削除されます。

  delcust で削除されるカスタマイズを確認するには、listcust に同じオプションを付けて使用します。delcust を使うとすべての管理カスタマイズが削除される可能性があるため、常に listcust を最初に実行して、意図どおりのものを削除しようとしていることを確認します。

delpg *name*
: 現在の選択内容のプロパティーグループ *name* を削除します。プロパティーグループを削除するために必要な特権については、smf_security(5) を参照してください。

  プロパティーグループがマニフェストまたはプロファイル内にある場合、そのプロパティーグループは非表示になります。smf(5) を参照してください。

delprop *pg*[/*name*]
: 現在の選択内容のプロパティーグループまたはプロパティーのうち、name で指定されているものを削除します。プロパティーを削除するために必要な特権については、smf_security(5) を参照してください。

delpropvalue *pg/name globpattern*
: 指定されたプロパティー値のうち、*glob*パターンに一致するものをすべて削除します。一致する値がなくても、成功します。プロパティーを変更するために必要な特権については、smf_security(5) を参照してください。

describe [-v] [-t] [*propertygroup*/*property*]
: 現在の設定、使用可能な設定のいずれかについて説明します。

  describe を引数なしで呼び出した場合、現在選択されているエンティティーおよび現在設定されているすべてのプロパティーグループやプロパティーの基本的な説明 (使用可能な場合) が表示されます。プロパティーグループや特定のプロパティーのクエリーを行うには、プロパティーグループ名か、プロパティーグループ名とプロパティー名をスラッシュ (/) で区切ったものを、引数として指定します。

-v オプションを指定すると、現在の設定、制約、使用可能なその他の設定オプションの説明など、使用可能なすべての情報が提供されます。

-t オプションを指定した場合は、選択内容のテンプレートデータのみが表示され (smf_template(5) を参照)、プロパティーグループやプロパティーの現在の設定は表示されません。

editprop
現在の選択内容のプロパティーグループおよびプロパティーを再生成するためのコマンドのコメントが一時ファイルに格納され、それを編集するために EDITOR 環境変数で指定されているプログラムが呼び出されます。完了すると、一時ファイル内のコマンドが実行されます。デフォルトのエディタは vi(1) です。プロパティーを作成、変更、または削除するために必要な特権については、smf_security(5) を参照してください。

listpg [*pattern*]
現在の選択内容のプロパティーグループの名前、タイプ、およびフラグを表示します。引数が指定されている場合、それは glob パターンと見なされ、引数に一致する名前を持つプロパティーグループだけが一覧表示されます。

対話モードでは、プロパティーグループの基本的な説明も提供されます。

listprop [-l *layer*...] [-f | -o *format*] [*pattern*]
現在の選択内容のプロパティーグループとプロパティーを一覧表示します。プロパティーグループの場合は、名前、タイプ、およびフラグが一覧表示されます。プロパティーの場合は、名前 (プロパティーグループ名とスラッシュ (/) が前に付加される)、タイプ、および値が一覧表示されます。使用可能なプロパティータイプのリストについては、scf_value_create(3SCF) を参照してください。引数が指定されている場合、それは glob パターンと見なされ、引数に一致する名前を持つプロパティーグループおよびプロパティーだけが一覧表示されます。

-l オプションを指定すると、値の定義元のレイヤーが出力されます。-l オプションにはレイヤーが必要であり、引数として manifest、system-profile、site-profile、admin、current、all を指定します。current は、-l を付けない listprop と同じプロパティー値を、値が定義されたレイヤーとともに出力します。

-f と -o オプションを同時に指定することはできません。-f は、プロパティーが定義されたファイルが存在する場合にそのファイルを出力します。-o ではフィールドの選択が可能です。 選択可能なフィールドは次のとおりです。

propname	プロパティー名
pgname	プロパティーグループ名
instname	インスタンス名
servicename	サービス名

`layer`	レイヤー
`proptype`	プロパティータイプ
`value`	プロパティー値
`file`	ソースファイル
`masked`	プロパティーグループまたはプロパティーが現在非表示であるかどうか
`time`	このプロパティーが最後に変更された時刻

`listcust` [`-L`] [`-M`] [*pattern*]

サイトプロファイルおよび管理レイヤーのカスタマイズと、現在選択されている非表示のエンティティーを出力します。引数が指定されている場合、それは glob パターンと見なされ、引数に一致する名前を持つプロパティーグループおよびプロパティーだけが一覧表示されます。現在の選択がない場合は、すべてのサーバーのすべてのカスタマイズが一覧表示されます。

`-M` が指定されている場合は、非表示のエンティティーのみが出力されます。

`-L` が指定されている場合は、すべてのローカルカスタマイズ (管理レイヤーとサイトプロファイルレイヤーの両方にあるものを含む) が表示されます。

`setenv` [`-i` | `-s`] [`-m` *method_name*] *envvar value*

method_name プロパティーグループのタイプが「method」の場合は、そのプロパティーグループの「environment」プロパティーを変更することにより、サービスまたはインスタンスのメソッドの環境変数を設定します。*method_name* が指定されていない場合で、`-i` オプションが使用されているときは、現在インスタンスが選択されていれば「method_context」プロパティーグループが使用されます。`-s` オプションが使用されている場合で、現在サービスが選択されているときは、その「method_context」プロパティーグループが使用されます。`-s` オプションが使用されている場合で、現在インスタンスが選択されているときは、その親の「method_context」プロパティーグループが使用されます。`-i` オプションと `-s` オプションのどちらも使用されていない場合は、現在選択されているエンティティー内で「start」プロパティーグループが検索されます。また、現在インスタンスが選択されている場合は、その親も検索されます。「inetd_start」プロパティーグループが見つからない場合は、同様の方法で検索されます。

プロパティーが見つかると、「*envvar=*」で始まる値がすべて削除され、値「*envvar=value*」が追加されます。プロパティーを変更するために必要な特権については、`smf_security(5)` を参照してください。

`setprop` *pg/name* = [*type:*] *value*

setprop *pg/name* = [*type:*] ([*values* ...])

現在の選択内容の *pg* プロパティーグループの *name* プロパティーを、タイプ *type* の指定された値に設定します。使用可能なプロパティータイプのリストについては、`scf_value_create(3SCF)` を参照してください。プロパティーがすでに存在し

ている場合は、そのプロパティーに既存の *type* が *type* と一致しないと、このサブコマンドは失敗します。値は二重引用符で囲まれる場合があります。二重引用符やバックスラッシュを含んでいる文字列値は、二重引用符で囲む必要があります。また、文字列値に含まれている二重引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。name で指定されたプロパティーが存在しない場合は、type が指定されていればプロパティーが作成されます。プロパティーを作成または変更するために必要な特権については、`smf_security(5)` を参照してください。複数の値は、指定された順番に格納されます。

`unsetenv` [`-i` | `-s`] [`-m` *method_name*] *envvar value*

method_name プロパティーグループのタイプが「method」の場合は、そのプロパティーグループの「environment」プロパティーを変更することにより、サービスまたはインスタンスのメソッドの環境変数を削除します。*method_name* が指定されていない場合で、`-i` オプションが使用されているときは、現在インスタンスが選択されていれば「method_context」プロパティーグループが使用されます。`-s` オプションが使用されている場合で、現在サービスが選択されているときは、その「method_context」プロパティーグループが使用されます。`-s` オプションが使用されている場合で、現在インスタンスが選択されているときは、その親の「method_context」プロパティーグループが使用されます。`-i` オプションと `-s` オプションのどちらも使用されていない場合は、現在選択されているエンティティー内で「start」プロパティーグループが検索されます。また、現在インスタンスが選択されている場合は、その親も検索されます。「inetd_start」プロパティーグループが見つからない場合は、同様の方法で検索されます。

プロパティーが見つかると、「*envvar=*」で始まる値がすべて削除されます。プロパティーを変更するために必要な特権については、`smf_security(5)` を参照してください。

通知パラメータのサブコマンド

`setnotify` {[`-g`] *tset* | *class*} *notification_parameters*

ソフトウェアイベントや障害管理問題ライフサイクルイベントの通知パラメータを、SMF リポジトリに設定します。

`-g`

SMF 状態遷移に対するシステム全体の通知パラメータを設定するために使用されます。`smf(5)` を参照してください。これらの通知パラメータは、`svccfg` の現在の選択内容にかかわらず `svc:/system/svc/global:default` 内に設定されます。このサブコマンドは、変更対象のインスタンスをすべて更新します。

class

FMA イベントのクラスまたは別名のコンマ区切りリスト。`smf(5)` の「通知パラメータ」を参照してください。

tset

SMF 状態遷移のコンマ区切りリスト。`smf(5)` の「通知パラメータ」を参照してください。

notification_parameters
実装されている各通知メカニズムの URI 形式。SMTP の場合は次を使用します。

`mailto:`*addr*`[?`*header1*`=`*value1*`[&`*header2*`=`*value2*`]]`

または

`mailto:{[active]|inactive}`

SNMP トラップの場合は次を使用します。

`snmp:{[active]|inactive}`

`smtp-notify(1M)` で定義されているパラメータ `msg_template` は、`mailto` URI のヘッダー値として設定できます。例:

`mailto:root@localhost?msg_template=<`*path to template file*`>`

SNMP トラップは、`/etc/net-snmp/snmp/snmpd.conf` 内の `trapsink` 指令で定義されたホスト、または SNMP トラップ通知デーモンで指定されたホストに送られます。`smtp-notify(1M)` を参照してください。

通知パラメータは、指定された class または tset に固有のものであり、事前に存在していた通知パラメータは上書きされます。active/inactive 形式は以前の通知パラメータを上書きしません。この形式では、指定された class または tset に対する通知メカニズムのオン/オフが切り替わるだけです。通知パラメータを設定すると、それらのパラメータが暗黙的にアクティブに設定されます。

`listnotify` [`-g`] [*tset*] | *class*
指定された class または tset の既存の通知パラメータを表示します。`-g` オプションを指定すると、`svc:/system/svc/global:default` の通知パラメータが表示されます。tset を省略すると、`all` が指定されたものとみなされます。

`delnotify` [`-g`] *tset* | *class*
指定された class または tset の既存の通知パラメータを削除します。`-g` オプションを指定すると、`svc:/system/svc/global:default` の通知パラメータが削除されます。

スナップショットのナビゲーションおよび選択のサブコマンド

`listsnap`
現在選択されているインスタンスで使用可能なスナップショットを表示します。

`revert` [*snapshot*]
現在選択されているインスタンスとそのサービスの管理カスタマイズを、指定されたスナップショットに記録されているプロパティーに戻します。引数が何も指定されていない場合は、現在選択されているスナップショットを使用し、成功時にそれを選択解除します。変更したプロパティー値は、`svcadm(1M)` の `refresh` サブコマンドを使用してアクティブにすることができます。プロパティーを変更するために必要な特権については、`smf_security(5)` を参照してください。

`selectsnap` [*name*]
現在のスナップショットを、*name* で指定されたスナップショットに変更します。*name* が指定されていない場合は、現在選択されているスナップショットを選択解除します。スナップショットは読み取り専用です。

インスタンスのサブコマンド

`refresh`
現在の構成に含まれる値を実行中のスナップショットに確定し、現在選択されているインスタンスがそれらの値を使用できるようにします。repository サブコマンドを使用せずにリポジトリが選択された場合、更新された構成を読み直すように、インスタンスのリスタータに指示します。

使用例

例 1 サービス記述のインポート

次の例では、コマンド行で指定した XML マニフェストに含まれる `seismic` サービスのサービス説明をインポートします。

```
# svccfg import /var/svc/manifest/site/seismic.xml
```

マニフェストは `service_bundle(4)` に指定されている形式に従う必要があります。

例 2 サービス説明のエクスポート

サービス説明をローカルシステムにエクスポートするには、次のように指定します。

```
# svccfg export dumpadm >/tmp/dump.xml
```

例 3 サービスインスタンスの削除

サービスインスタンスを削除するには、次のように指定します。

```
# svccfg delete network/inetd-upgrade:default
```

例 4 代替リポジトリに含まれるプロパティーの検査

代替リポジトリをロードしたあとでサービスのプロパティーの状態を調べるには、次に示す一連のコマンドを使用します。このようなコマンドを使用して、たとえば、特定のリポジトリのバックアップ内でサービスが有効になっていたかどうかを調べることができます。

```
# svccfg
svc:> repository /etc/svc/repository-boot
svc:> select telnet:default
svc:/network/telnet:default> listprop general/enabled
general/enabled  boolean  false
svc:/network/telnet:default> exit
```

例 5 デバッグの有効化

start メソッドの `LD_PRELOAD` を変更し、デバッグ機能が有効になっている `libumem(3LIB)` を使用できるようにするには、次のように指定します。

例 5 デバッグの有効化 *(続き)*

```
$ svccfg -s system/service setenv LD_PRELOAD libumem.so
$ svccfg -s system/service setenv UMEM_DEBUG default
```

例 6 describe サブコマンドの使用

次のコマンドは、describe サブコマンドの使用法を示しています。

```
# svccfg -s console-login describe ttymon
ttymon                      application
ttymon/device               astring  /dev/console
   terminal device to be used for the console login prompt
ttymon/label                astring  console
   appropriate entry from /etc/ttydefs
   ...
```

例 7 通知設定の構成

次のコマンドは、SMF サービス状態遷移イベントの通知パラメータを構成します。

```
# svccfg setnotify -g from-online,to-maintenance \
mailto:admin@somehost.com
```

例 8 SNMP 通知の有効化

次のコマンドは、障害管理イベントの SNMP 通知を有効にします。

```
# svccfg setnotify problem-diagnosed,problem-updated \
mailto:admin@somehost.com snmp:
```

例 9 通知設定の一覧表示

次のコマンドは、障害管理イベントの通知設定を一覧表示します。

```
# svccfg listnotify problem-diagnosed,problem-updated
Event: problem-diagnosed
  Notification Type: smtp
    active: true
   to: admin@somehost.com
  Notification Type: snmp
    active: true

Event: problem-updated
  Notification Type: smtp
    active: true
   to: admin@somehost.com
  Notification Type: snmp
    active: true
```

環境変数

EDITOR

editprop サブコマンドが使用されたときに実行するコマンド。デフォルトのエディタは vi(1) です。

終了ステータス

次の終了値が返されます。

0

正常終了。

1

1 つまたは複数のサブコマンドが失敗しました。エラーメッセージが標準エラーストリームに書き込まれます。

2

無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照。

対話型出力は「不確実」です。呼び出しおよび非対話型出力は「確実」です。

関連項目

svcprop(1), svcs(1), smtp-notify(1M), svcadm(1M), svc.configd(1M), libscf(3LIB), libumem(3LIB), scf_service_add_pg(3SCF), scf_value_create(3SCF), contract(4), service_bundle(4), attributes(5), fnmatch(5), smf(5), smf_method(5), smf_security(5), smf_template(5)

名前

svc.configd – サービス管理機能のリポジトリデーモン

形式

/lib/svc/bin/svc.configd

機能説明

svc.configd は、サービス管理機能のリポジトリデーモンです。svc.configd はシステムの起動中に自動的に起動され、なんらかの障害が発生した場合は再起動されます。svc.configd を直接起動してはいけません。

svc.configd との対話は、libscf(3LIB)、およびコマンド行ツール (svcs(1)、svcprop(1)、svcadm(1M)、および svccfg(1M)) を介して行われます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

svcs(1), svcprop(1), svcadm(1M), svccfg(1M), libscf(3LIB), attributes(5)

名前

svc.ipfd – IP フィルタファイアウォール監視デーモン

形式

```
/lib/svc/bin/svc.ipfd

svc:/network/ipfilter:default
```

機能説明

svc.ipfd デーモンは、ファイアウォール構成を使用するサービスに対するアクションを監視し、更新サービスの IP フィルタ構成を開始します。このデーモンにより、システムのファイアウォール構成の変更に対して、システムがサービスレベルごとに増分方式で対応できるようになります。

サービスのファイアウォールポリシーは、有効にされたときにはアクティブ化、無効にされたときには非アクティブ化され、その構成プロパティーグループが変更されたときに更新されます。svc.ipfd は、これらのアクションのサービス管理機能 (SMF) リポジトリを監視し、サービスのファイアウォールポリシーを実行するための IP フィルタ規則生成プロセスを起動します。

このデーモンは、start メソッドまたは refresh メソッドのどちらかを介して、network/ipfilter サービスによって起動されます。そのため、このデーモンはそのメソッドから環境変数と資格情報を継承し、すべてのゾーン特権を持つルートとして実行されます。

ファイアウォールの静的な構成

静的な定義には、サービス固有の IPF 規則を生成するために使用される、そのサービスのネットワークリソース構成が記述されます。サービスごとの firewall_context プロパティーグループには、inetd 管理対象サービス内の inetd プロパティーグループと同様に、サービスの静的な定義が含まれます。このプロパティーグループは、次の値をサポートしています。

firewall_context/name
: inetd 以外のサービス用。inetd/name プロパティーと同等の IANA 名または RPC 名です。

firewall_context/isrpc
: inetd 以外のサービス用。inetd/isrpc プロパティーと同等のブール型プロパティー。ここで、true 値は RPC サービスを示します。RPC サービスの場合、firewall_context/name の値は IANA 名ではなく、RPC プログラム番号または名前のどちらかです。rpc(4) を参照してください。

さらに、一部のサービスでは、独自の IPF 規則を生成して提供するためのメカニズムが必要になる場合があります。オプションのプロパティー ipf_method は、このようなカスタム規則の生成を可能にするためのメカニズムを提供します。

firewall_context/ipf_method
: コマンド。通常は、サービスの IPF 規則を生成するスクリプトです。このフレームワークでは、このプロパティー定義を含むサービスの規則は生成しません。代わりに、これらのサービスが独自の規則を提供するものと想定します。

サービスの ipf_method は、追加の引数である独自の障害管理リソース識別子 (FMRI) を受け取り、そのサービスのファイアウォール規則を生成して標準出力に

出力するコマンドを指定します。このフレームワークでは、ipf_method プロパティーを備えたサービスの規則を生成するために、ipf_method で指定された (そのサービスの FMRI を追加の引数として渡す) コマンドを実行し、コマンド出力 (その規則) をサービスの規則ファイルにリダイレクトすることによってそのサービスの規則を格納します。ipf_method は、network/ipfilter start または refresh メソッドプロセスのどちらかのコンテキストから exec で実行されるため、実行コンテキストを継承し、ルートとして実行されます。

サービスの静的な構成はサービス開発者によって提供され、ユーザーによる変更は意図されていません。これらのプロパティーは、更新されたサービス定義のインストール時にのみ変更されます。

ファイアウォールポリシーの構成

サービスごとのプロパティーグループ firewall_config は、そのサービスのファイアウォールポリシー構成を格納します。network/ipfilter:default は、(ipfilter(5) で説明しているように) システム全体のポリシーであるグローバルデフォルトとグローバルオーバーライドの 2 つのファイアウォールポリシーを担当しているため、それぞれのシステム全体のポリシーを格納するために firewall_config_default と firewall_config_override の 2 つのプロパティーグループを備えています。

各プロパティー、指定できる値、および対応するセマンティクスは次のとおりです。

policy
: policy には、次のモードがあります。

  none ポリシーモード
  : アクセス制限はありません。グローバルポリシーの場合、このモードでは、すべての受信トラフィックを許可します。サービスポリシーの場合、このモードでは、そのサービスへのすべての受信トラフィックを許可します。

  deny ポリシーモード
  : none より制限的なモードです。このモードでは、apply_to プロパティーで指定された発信元を除くすべての発信元からの受信トラフィックを許可します。

  allow ポリシーモード
  : もっとも制限的なモードです。このモードでは、apply_to プロパティーで指定された発信元を除くすべての発信元からの受信トラフィックをブロックします。

apply_to
: 選択されたポリシーモードを適用するネットワークエンティティーの一覧が示された複数値プロパティー。apply_to プロパティーに示されているエンティティーは、ポリシーが deny の場合は拒否され、ポリシーが allow の場合は許可されます。指定可能な値の構文は次のとおりです。

host:	host:*IP*	"host:192.168.84.14"
subnet:	network:*IP/netmask*	"network:129.168.1.5/24"
ippool:	pool:*pool number*	"pool:77"
interface:	if:*interface_name*	"if:e1000g0"

exceptions
: apply_to リストから除外されるネットワークエンティティーの一覧が示された複数値プロパティー。たとえば、あるサブネットに deny ポリシーが適用されている場合、そのサブネット内の一部のホストを exceptions プロパティーで指定することによって、それらのホストに例外を設定できます。このプロパティーには、apply_to プロパティーと同じ値の構文があります。

個別のネットワークサービスのみの場合

firewall_config/policy
: サービスのポリシーを use_global に設定することもできます。use_global ポリシーモードのサービスは、グローバルデフォルトのファイアウォールポリシーを継承します。

グローバルデフォルトのみの場合

firewall_config_default/policy
: svc:/network/ipfilter:default 内のグローバルデフォルトポリシー firewall_config プロパティーグループを custom に設定することもできます。ユーザーは、policy を custom に設定することで、このフレームワークで提供できない既存の IP フィルタ構成やカスタム構成など、事前に設定された IP フィルタ構成を使用できます。このグローバルデフォルトのみのポリシーモードでは、ユーザーは、IPF 規則の完全なセットを含むテキストファイルを設定できます。custom モードが選択されている場合、IPF 規則の指定されたセットは「完全」であり、このフレームワークは構成済みのファイアウォールポリシーから IPF 規則を生成しません。

firewall_config_default/custom_policy_file
: グローバルデフォルトポリシーが custom に設定されているときに使用されるファイルパス。このファイルには、目的の IP フィルタ構成を提供する IPF 規則のセットが含まれます。たとえば、/etc/ipf/ipf.conf 内に既存の IPF 規則がある場合、ユーザーは次のコマンドを実行してその既存の規則を使用できます。

1. カスタムポリシーを設定します。

```
# svccfg -s ipfilter:default setprop \
firewall_config_default/policy = astring: "custom"
```

2. カスタムファイルを指定します。

```
# svccfg -s ipfilter:default setprop \
firewall_config_default/custom_policy_file = astring: \
"/etc/ipf/ipf.conf"
```

3. 構成を更新します。

```
# svcadm refresh ipfilter:default
```

firewall_config_default/open_ports
: 受信トラフィックを許可する必要があるサービス以外のプログラムは、ファイアウォールがその通信ポートへのトラフィックを許可するよう要求できます。この複数値プロパティーには、プロトコルとポートの組が次の形式で含まれます。

"{tcp | udp}:{*PORT* | *PORT-PORT*}"

最初に、システム全体のポリシーが none に設定され、ネットワークサービスのポリシーが use_global に設定されます。システム全体のポリシーが none で、すべてのサービスがそのポリシーを継承するため、network/ipfilter を有効にすると、IP フィルタ規則の空のセットを備えたファイアウォールが有効になります。より制限的なポリシーを構成するには、svccfg(1M) を使用して、ネットワークサービスとシステム全体のポリシーを変更します。

ユーザーは、サービスの firewall_config プロパティーグループを変更することによってファイアウォールポリシーを構成します。ファイアウォールの管理者特権のユーザーに委譲できるようにするため、新しい承認である solaris.smf.value.firewall.config が作成されます。サービスオペレータの特権を持つユーザーがファイアウォールポリシーを構成できるようにするには、この新しい承認が必要です。

ファイアウォールの使用条件

ブート中、有効になっているサービスが起動する前に、これらのサービスのためにファイアウォールが構成されます。そのため、ブート時にはサービスが保護されます。システムの稼働中は、サービスの再起動、有効化、更新などの管理操作により、ファイアウォールが構成されている間に実行されるサービスが短時間脆弱な状態になる可能性があります。

svc.ipfd は、サービスの起動イベントと停止イベントを監視し、SMF がそのサービスを起動または停止するのと同時にサービスのファイアウォールを構成または構成解除します。この 2 つの操作は同時に行われるため、ファイアウォールの構成が完了する前にサービスが起動された場合は、危険にさらされやすい期間が (1 秒未満) 発生する可能性があります。通常、RPC サービスは一時アドレス上で待機しますが、このアドレスはサービスが実際に実行されるまで不明です。したがって、サービスが実行されるまでファイアウォールが構成されないため、RPC サービスは同じように危険にさらされやすい状態になります。

開発者向けドキュメント

リモート機能を提供するサービスは、そのサービスへのネットワークアクセスを制御するために、ファイアウォールフレームワークに参加させることをお勧めします。フレームワークの統合は必須ではありませんが、システム全体のポリシーが構成されている場合、このフレームワークに統合されていないサービスへのリモートアクセスは正しく機能しない可能性があります。

このフレームワークへのサービスの統合は、2 つの追加のプロパティーグループとそれに対応するプロパティーをサービスマニフェスト内で定義するだけで完了します。IP フィルタ規則は、ユーザーがそのサービスを有効にしたときに生成されま

す。カスタム規則を生成するという重要な状況では、シェルスクリプトが必要になりますが、例として使用できるスクリプトがあらかじめ用意されています。

追加のプロパティーグループである firewall_config と firewall_context はそれぞれ、ファイアウォールポリシーの構成の格納と、静的なファイアウォール定義の提供を行います。新しいプロパティーグループとプロパティー、およびそれに対応するデフォルト値の要約を次に示します。

ファイアウォールポリシーの構成

firewall_config
: システムへのアクセスは、新しい承認定義とユーザー定義のプロパティータイプによって保護されます。この新しい承認は、次のような方法で、このプロパティーグループの value_authorization プロパティーに割り当てるようにしてください。

```
<propval name='value_authorization' type='astring'
value='solaris.smf.value.firewall.config' />
```

サードパーティーは、ユーザー定義タイプを生成するために、サービスシンボル名前空間の規則に従うようにしてください。Sun が提供するサービスでは、com.sun,fw_configuration をプロパティータイプとして使用できます。

詳細については、前述の「ファイアウォールポリシーの構成」を参照してください。

firewall_config/policy
: サービスは、デフォルトではグローバルデフォルトのファイアウォールポリシーを継承するため、このプロパティーの初期値は use_global にするようにしてください。

firewall_config/apply_to
: 空のプロパティー。このプロパティーに初期値はありません。

firewall_config/exceptions
: 空のプロパティー。このプロパティーに初期値はありません。

ファイアウォールの静的な定義

firewall_context
: サードパーティーは、ユーザー定義のタイプを生成するために、サービスシンボル名前空間規則に従うようにしてください。Sun が提供するサービスでは、com.sun,fw_definition をプロパティータイプとして使用できます。

詳細については、前述の「ファイアウォールの静的な構成」を参照してください。

`firewall_context/name`
: IANA で定義された既知のポートを備えたサービスで、getservbyname(3SOCKET) によって取得できます。このプロパティーには、このサービスの IANA 名が格納されています。RPC サービスの場合、このプロパティーには RPC プログラム番号が格納されています。

`firewall_context/isrpc`
: RPC サービスの場合、このプロパティーは値を `true` に設定して作成するようにしてください。

`firewall_context/ipf_method`
: 一般に、指定されたファイアウォールポリシーは、`firewall_context/name` プロパティーから得られる、サービスの通信ポートに対する IP フィルタ規則を生成するために使用されます。IANA で定義されたポートを備えておらず、RPC サービスではないサービスは、独自の IP フィルタ規則を生成する必要があります。独自の規則を生成するサービスは、`firewall_context/name` プロパティーと `firewall_context/isrpc` プロパティーを持たないようにできます。次のサービスを参照してください

```
svc:/network/ftp:default
svc:/network/nfs/server:default
svc:/network/ntp:default
```

そのほかに、`ipf_method` を備えたものもガイダンスとして参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os, network/ipfilter
インタフェースの安定性	確実

関連項目

svcprop(1), svcs(1), ipf(1M), svcadm(1M), svccfg(1M), getservbyname(3SOCKET), rpc(4), attributes(5), ipfilter(5), smf(5)

名前

svc.startd – サービス管理機能のマスターリスタータ

形式

/lib/svc/bin/svc.startd

svc:/system/svc/restarter:default

機能説明

svc.startd はサービス管理機能 (SMF) のマスターリスタータデーモンであり、すべてのサービスのデフォルトリスタータです。svc.startd は、管理要求、システム障害、またはアプリケーション障害に基づいて、サービスの開始、停止、および再起動を行います。

svc.startd は、各サービスの依存関係に基づいて障害管理を担うほか、サービスの状態の維持も行います。

svc.startd はシステムの起動時に自動的に呼び出されます。なんらかの障害が発生した場合は再起動されます。直接 svc.startd を呼び出さないでください。

すべてのリスタータに共通する構成および動作の詳細については、smf_restarter(5) を参照してください。

svcs(1) は、サービス構成機能によって管理されているすべてのサービスについてステータスを報告します。svcadm(1M) を使用すると、サービスのリスタータに関連してサービスインスタンスを操作できます。

環境変数

「SMF_」という接頭辞を持つ環境変数は予約済みであり、上書きが可能です。

svc.startd は、smf_method(5) で指定されている「SMF_」環境変数をメソッドに渡します。デフォルトでは、PATH は /usr/sbin:/usr/bin に設定されます。デフォルトでは、svc.startd に渡されるその他すべての環境変数は、init(1M) から継承されたものです。

重複するエントリは削減されて単一のエントリになります。使用される値は不定です。「<名前>=」という接頭辞を持たない環境エントリは無視されます。

リスタータオプション

svc.startd の構成はコマンド行オプションでは設定されません。代わりに、サービス構成リポジトリから構成が読み込まれます。svccfg(1M) を使用すると、すべてのオプションおよびプロパティーを設定できます。

次に示す options プロパティーグループ内の構成変数は、開発者および管理者が使用できます。

boot_messages
: *astring* は、ブート中にコンソールに出力するメッセージのデフォルトレベルを示します (scf_value_is_type で定義される。scf_value_create(3SCF) を参照)。サポートされているメッセージオプションには、quiet と verbose があります。quiet オプションは、ブート中に最小限のメッセージをコンソールに出力します。verbose オプションは、開始されるサービスごとに 1 つのメッセージを出力して、成功または失敗を示します。boot -m オプションを使用すると、ブート時に boot_messages 設定を上書きできます。kernel(1M) を参照してください。

logging
: svc.startd のグローバルサービスログ記録レベルを制御します。*astring* (scf_value_is_type で定義される。scf_value_create(3SCF) を参照) は、syslog (syslog(3C) を参照) および svc.startd のグローバルログファイル /var/svc/log/svc.startd.log に記録するメッセージのデフォルトレベルを示します。サポートされているメッセージオプションには、quiet、verbose、および debug があります。quiet オプションは、管理者の介入を必要とするエラーメッセージを、コンソール、syslog、および svc.startd のグローバルログファイルに送信します。verbose オプションは、管理者の介入を必要とするエラーメッセージをコンソール、syslog、および svc.startd のグローバルログファイルに、管理者の介入を必要としないエラーに関する情報を svc.startd のグローバルログファイルに、それぞれ送信します。さらに、サービスが起動されるたびに単一のメッセージがコンソールに送信されます。debug オプションは、svc.startd デバッグメッセージを svc.startd のグローバルログファイルに、管理者の介入を必要とするエラーメッセージをコンソール、syslog、および svc.startd のグローバルログファイルに、サービスが起動されるたびに単一のメッセージをコンソールに、それぞれ送信します。

milestone
: デフォルトのブートレベルとして使用するマイルストーンを決定する FMRI です。指定できるオプションは、次に示すメジャーマイルストーンです。

```
svc:/milestone/single-user:default
svc:/milestone/multi-user:default
svc:/milestone/multi-user-server:default
```

あるいは特殊値 all または none も指定できます。all は、すべてのサービスに依存する理想化されたマイルストーンを表します。none は、マスター svc:/system/svc/restarter:default 以外にはどのサービスも実行されていない、特殊なマイルストーンです。デフォルトでは、svc.startd は、すべてのサービスに依存する統合的なマイルストーン all を使用します。このプロパティーが指定された場合、inittab(4) の initdefault 設定はすべて上書きされます。

system/reconfigure
: 再構成リブートが要求されたことを示します。再構成リブートに応じて動作する必要のあるサービスでは、このプロパティーが存在し 1 に設定されているかどうかをチェックして、再構成ブートが要求されたかどうかを確認できます。

このプロパティーは svc.startd によって管理されます。管理者はこのプロパティーを変更しないでください。

svc.startd を無効にするなどの構成エラーは、syslog によってログに記録されますが、無視されます。

サービスの状態

svc.startd によって管理されるサービスは、smf(5) で説明されている状態のいずれかになります。このリスタータによって状態の定義が変更されることはありません。

サービスの報告機能

管理対象サービスによって実行されるログ記録のほかに、svc.startd には、サービスの報告およびログ記録を行う共通のメカニズムが用意されています。

報告機能プロパティー svc.startd は、管理しているすべてのサービスについて、共通のプロパティーセットを更新します。これらのプロパティーは、サービスインスタンスの状態に基づいてアクションを実行するための共通インタフェースとして使用できます。svcs(1) コマンドを使用すると、これらのプロパティーを簡単に表示できます。

restarter/state
restarter/next_state
: インスタンスの現在の状態および次の状態 (移行中の場合)。

restarter/auxiliary_state
: 現在のインスタンス状態について詳細情報を示す説明文。svc.startd によって管理されるサービスには、次に示す補助状態を使用できます。

maintenance

fault_threshold_reached
stop_method_failed
administrative_request

restarter/state_timestamp
: 現在の状態に到達した時刻。

restarter/contract
: サービスインスタンスを実行している主プロセス契約 ID (存在する場合)。

ログ

デフォルトでは、svc.startd は、メソッドの標準出力および標準エラーのファイル記述子に加えて、サービスに対する重要なリスタータアクションを /var/svc/log/*service*:*instance*.log に記録します。/var/svc/log/svc.startd.log や syslog など、システムのグローバルな場所へのロギングレベルは、options/logging プロパティーによって制御されます。

サービスの定義

svc.startd によって管理されるサービスを開発または構成するとき、サービスインスタンスとリスタータの相互作用を操作するために、一連の共通プロパティーが使用されます。

メソッド

svc.startd で提供されている fork/exec モデル用メソッドの一般的な形式は、smf_method(5) に示されています。svc.startd によって管理されるサービスでは、次に示すメソッドが必須メソッドまたは省略可能メソッドとしてサポートされています。

refresh
: サービスを中断することなく、リポジトリまたは config ファイルから適切な構成パラメータを読み込み直します。多くの場合、これはシステムデーモン用の SIGHUP を使用して実装されます。再起動しないとサービスが構成の変更を認識できない場合、refresh メソッドは提供されません。

  このメソッドは省略可能です。

start
: サービスを開始します。アプリケーションがコンシューマに対して利用可能になったときに初めて成功を返します。競合するインスタンスがすでに実行されている場合や、サービスを開始できない場合は、失敗を返します。

  このメソッドは必須です。

stop
: サービスを停止します。場合によっては、一部またはすべてのサービスがすでに停止されているときに stop メソッドを呼び出すこともできます。メソッドが戻るときにサービスが完全に停止していない場合だけ、エラーを返します。

  このメソッドは必須です。

サービスが必須メソッド内で何もアクションを実行する必要がない場合、そのメソッドに :true トークンを指定する必要があります。

svc.startd は、サービスや特定のメソッドに対して指定されたメソッドコンテキストをすべて尊重します。smf_method(5) で説明されているメソッド拡張トークンは、svc.startd によって呼び出されるすべてのメソッドで使用できます。

プロパティー

smf(5) で使用できる一般的なプロパティーの概要。これらの一般的なプロパティーは、次のような特定の方法で svc.startd と対話します。

general/enabled
: enabled が true に設定されている場合、リスタータは、サービスの依存関係がすべて満たされるとサービスの開始を試みます。false に設定されている場合、サービスは実行されず、無効にされた状態のままになります。

`general/restarter`
: この FMRI プロパティーが空または `svc:/system/svc/restarter:default` に設定されている場合、サービスは `svc.startd` によって管理されます。それ以外の場合は、指定されたリスタータが (利用可能になったときに) サービスの管理を担当します。

`general/single_instance`
: `single_instance` が true に設定されている場合、`svc.startd` は、常にこのサービスの 1 つのインスタンスのみ、オンラインまたは機能低下状態に移行することを許可します。

また、`svc.startd` によって管理されるサービスは、下記に示す省略可能プロパティーを `startd` プロパティーグループ内に定義できます。

`startd/duration`
: `duration` プロパティーは、サービスのモデルを定義します。このプロパティーは、`transient`、`wait` とも呼ばれる `child` モデルサービス、または `contract` (デフォルト) に設定できます。

`startd/ignore_error`
: `ignore_error` プロパティーは、設定されている場合、無視するイベントをコンマ区切りのリストで指定します。このリストで有効な文字列値は `core` と `signal` です。デフォルトでは、すべてのエラーで再起動が行われます。

`startd/need_session`
: `need_session` プロパティーが true に設定されている場合は、インスタンスを独自のセッションで起動する必要があることを示します。デフォルトでは、そのようにはなりません。

`startd/utmpx_prefix`
: `utmpx_prefix` 文字列プロパティーは、メソッドを実行する前にインスタンスに有効な `utmpx` エントリが必要であることを定義します。デフォルトでは、`utmpx` エントリは作成されません。

サービス障害

メソッドからゼロ以外の終了コードが返された場合や指定されたタイムアウトの期限が切れる前にメソッドが完了できなかった場合、`svc.startd` はメソッドが失敗したと見なします。`$SMF_EXIT_ERR_CONFIG` または `$SMF_EXIT_ERR_FATAL` が返された場合、`svc.startd` はすぐにサービスを保守状態にします。その他すべての障害では、`svc.startd` はサービスをオフライン状態にします。サービスがオフラインの場合で、その依存関係が満たされているときは、`svc.startd` は再びサービスの開始を試みます (`smf(5)` を参照)。

contract または transient サービスが、その定義済みのタイムアウトが経過する前に起動メソッドから戻らない場合、`svc.startd` はそのメソッドに `SIGKILL` を送信し、サービスをオフライン状態に戻します。

障害が 5 回連続して発生した場合、またはエラーのためにサービスが 10 分経過する間に複数回再起動されている場合、`svc.startd` はサービスを保守状態にします。

サービス障害の条件は、startd/duration プロパティーで定義されるサービスモデルと、startd/ignore_error プロパティーの値との組み合わせによって定義されます。

次に示す条件のいずれかが発生すると、contract モデルサービスは障害状態になります。

- サービスのすべてのプロセスが終了する
- サービスのプロセスのいずれかでコアダンプが生成される
- サービス外部のプロセスからサービスプロセスに致命的な信号が送信される (たとえば、管理者が pkill コマンドを使用してサービスプロセスを終了する)

startd/ignore_error に core または signal、あるいはその両方を指定すると、サービスでは最後の 2 つの条件を無視できます。

サービスを transient として定義すると、svc.startd はそのサービスのプロセスを追跡しません。したがって、contract モデルサービスについて記述されている潜在的な障害は、transient サービスについては障害と見なされません。transient サービスは、メソッド失敗条件のいずれかが発生した場合だけ保守状態になります。

wait モデルサービスは、サービスに関連付けられている子プロセスが終了するたびに再起動されます。「wait」モデルサービスの場合、子プロセスの終了はエラーとは見なされず、障害が繰り返し発生しても保守状態への移行にはつながりません。

レガシーサービス

svc.startd は、起動時の実行レベル移行中に呼び出されるサービスを引き続きサポートしています。同等の実行レベルのマイルストーンを構成しているすべての管理対象サービスがオンライン状態に移行したあと、各 /etc/rc?.d ディレクトリが処理されます。/etc/rc?.d ディレクトリにある標準の init スクリプトは、そのシーケンス番号の順に実行されます。

マイルストーンと実行レベルのマッピングは次のとおりです。

milestone/single-user	シングルユーザー (S)
milestone/multi-user	マルチユーザー (2)
milestone/multi-user-server	ネットワークサービス付きマルチユーザー (3)

また、svc.startd は、スクリプトごとに 1 つのインスタンスをリポジトリに挿入することにより、これらのレガシーサービスが SMF で表示されるようにします。これらのレガシーインスタンスは、svcs(1) などの標準の SMF インタフェースを使用して表示でき、常に LEGACY-RUN 状態で表示されます。変更したり、ほかのサービスの依存関係として指定したりすることはできません。管理者に役立つように、レガシーサービスの初期の開始時刻が取り込まれます。

ファイル

/var/svc/log	svc.startd がログファイルを格納するディレクトリ。
/etc/svc/volatile	/var が読み書き用にマウントされる前のブートの初期段階で svc.startd がログファイルを格納するディレクトリ。

使用例

例 1　詳細ログをオンに設定する

詳細ログをオンに設定するには、次のように入力します。

```
# /usr/sbin/svccfg -s system/svc/restarter:default
svc:/system/svc/restarter:default> addpg options application
svc:/system/svc/restarter:default> setprop options/logging = \
astring: verbose
svc:/system/svc/restarter:default> exit
```

この要求は、次に svc.startd を再起動したときに有効になります。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

svcs(1), svcprop(1), kernel(1M), init(1M), svcadm(1M), svccfg(1M), svc.configd(1M), setsid(2), syslog(3C), libscf(3LIB), scf_value_create(3SCF), contract(4), init.d(4), process(4), inittab(4), attributes(5), smf(5), smf_method(5)

名前

swap – スワップ管理インタフェース

形式

```
/usr/sbin/swap -a swapname [swaplow] [swaplen]

/usr/sbin/swap -d swapname [swaplow]

/usr/sbin/swap -l [-h | -k]

/usr/sbin/swap -s [-h]
```

機能説明

swap ユーティリティーは、メモリーマネージャーが使用するシステムスワップ領域を追加、削除、および監視する方法を提供します。

オプション

次のオプションがサポートされています。

-a *swapname* [*swaplow*] [*swaplen*]
指定されたスワップ領域を追加します。このオプションを使用できるのは、スーパーユーザーか、Primary Administrator 役割を引き受けているユーザーのみです。*swapname* はスワップ領域または通常ファイルの名前です。たとえば、UFS ルートファイルシステムを実行しているシステムでは、/dev/dsk/c0t0d0s1 などのスライスか、通常ファイルをスワップ領域として指定します。ZFS ファイルシステムを実行しているシステムでは、/dev/zvol/dsk/rpool/swap などの ZFS ボリュームをスワップ領域として指定します。ZFS ファイルシステムでは、スワップに通常ファイルを使用することはできません。また、システムが ZFS ルートファイルシステムを実行しているときに、スワップ領域とダンプデバイスの両方に同じ ZFS ボリュームを使用することはできません。

swaplow は、ファイル内でスワップ領域を開始する場所のオフセット (512 バイトブロック単位) です。*swaplen* は、スワップ領域の望ましい長さ (512 バイトブロック単位) です。*swaplen* の値を 16 より小さくすることはできません。たとえば、*n* ブロックが指定された場合は、(*n*-1) ブロックが実際のスワップの長さになります。*swaplen* は、少なくとも 1 ページの長さにする必要があります。メモリーのページのサイズは、pagesize コマンドを使用して決定できます。pagesize(1) を参照してください。スワップファイルの最初のページは自動的にスキップされ、スワップファイルは少なくとも 1 ページの長さにする必要があるため、最小サイズは 2 ページサイズバイトの倍数にしてください。メモリーのページのサイズは、マシンに依存します。

swaplow + *swaplen* は、スワップファイルのサイズ以下にする必要があります。*swaplen* が指定されていない場合は、*swaplow* から始まり、指定されたファイルの最後まで拡張された領域が追加されます。*swaplow* も *swaplen* も指定されていない場合は、最初のページを除く、ファイル全体が使用されます。スワップ領域は通常、システムの起動中に /usr/sbin/swapadd スクリプトによって自動的に追加されます。このスクリプトは、/etc/vfstab ファイルで指定されたすべてのスワップ領域を追加します。これらの指定の構文については、vfstab(4) を参照してください。

vfstab(4) で encrypted オプションを指定し、ZFS ボリュームに encryption プロパティーを指定すると、スワップデバイスとして使用される ZFS ボリュームを暗号化できます。 zfs(1M) を参照してください。

NFS またはローカルファイルシステム *swapname* を使用するには、最初に mkfile(1M) を使用してファイルを作成することをお勧めします。ローカルファイルシステムのスワップファイルの場合は、swap -a コマンドを実行するだけで、稼働中のシステムに追加できます。NFS マウントされたスワップファイルの場合は、サーバーがファイルをエクスポートする必要があります。これを行うには、次の手順を実行します。

1. /etc/dfs/dfstab に次の行を追加します。

   share -F nfs -o \
   rw=*clientname*,root=*clientname path-to-swap-file*

2. shareall(1M) を実行します。
3. クライアントに、/etc/vfstab に次の行を追加させます。

   server:*path-to-swap-file* - *local-path-to-swap-file* nfs \
   - - - *local-path-to-swap-file* - - swap - - -

4. クライアントに mount を実行させます。

   # mount *local-path-to-swap-file*

5. クライアントは次に、swap -a を実行してスワップ空間を追加できます。

   # swap -a *local-path-to-swap-file*

-d *swapname*
指定されたスワップ領域を削除します。このオプションは、スーパーユーザーのみが使用できます。*swapname* は、スワップファイルの名前です。たとえば、/dev/dsk/c0t0d0s1 または通常ファイルです。*swaplow* は、削除されるスワップ領域のオフセット (512 バイトブロック単位) です。*swaplow* が指定されていない場合は、第 2 ページから領域の削除が開始されます。このコマンドが完了すると、この領域からスワップブロックを割り当てることはできなくなり、以前このスワップ領域で使用されていたすべてのスワップブロックがほかのスワップ領域に移動されています。

-h
すべてのサイズが人間の読める形式にスケーリングされます。縮小拡大/スケーリングは、1024 で繰り返し除算することによって行われます。

-k
ファイルサイズを 1024 バイトの単位で出力します。

-l
すべてのスワップ領域のステータスを一覧表示します。出力には次の 5 つの列があります。

`path`
スワップ領域のパス名。

`dev`
ブロック特殊デバイスの場合は、メジャー/マイナーデバイス番号 (10 進数)。それ以外の場合は 0。

`swaplo`
領域の *swaplow* 値 (512 バイトブロック単位)。

`blocks`
領域の *swaplen* 値 (512 バイトブロック単位)。

`free`
この領域内で現在割り当てられていない 512 バイトブロックの数。

この空間は特定のスワップ領域には関連付けられていないため、このリストには物理メモリーの形式でのスワップ空間は含まれません。

swapname の (`swap -d` による) 削除処理中に `swap -l` を実行した場合は、スワップ状態の 6 番目の列に文字列 INDEL が表示されます。

`-s`
全体のスワップ空間の使用量および利用可能量に関する要約情報を出力します。

`allocated`
バッキングストアとして使用するように現在割り当てられているスワップ空間の合計容量 (バイト単位)。

`reserved`
現在は割り当てられていないが、あとから使用できるようにメモリーマッピングによって確保されているスワップ空間の合計容量 (バイト単位)。

`used`
割り当て済みまたは予約済みのスワップ空間の合計容量 (バイト単位)。

`available`
あとから予約や割り当てに使用可能な合計スワップ空間 (バイト単位)。

これらの数値には、`-l` オプションによって表示されるすべての構成済みスワップ領域のスワップ空間や、物理メモリーの形式でのスワップ空間が含まれます。

使用法

32 ビットオペレーティングシステムでは、2G バイト以上のサイズのスワップデバイスには最初の 2G バイト -1 のみが使用されます。64 ビットオペレーティングシステムでは、2G バイトを超えるブロックデバイスを最大 2^{63} –1 バイトまでのスワップに完全に使用できます。

環境変数

swap の実行に影響を与える次の環境変数についての詳細は、environ(5) を参照してください。LC_CTYPE および LC_MESSAGE。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

pagesize(1), mkfile(1M), shareall(1M), zfs(1M), getpagesize(3C), vfstab(4), attributes(5), largefile(5)

注意事項

ZFS でのスワップ領域の設定については、『Solaris ZFS 管理ガイド』を参照してください。

警告

追加されるスワップ領域が既存のファイルシステムとオーバーラップするかどうかかの確認は行われません。

名前

sync – update the super block

形式

sync

機能説明

sync executes the sync system primitive. If the system is to be stopped, sync must be called to ensure file system integrity. It will flush all previously unwritten system buffers out to disk, thus assuring that all file modifications up to that point will be saved. See sync(2) for details.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

sync(2), attributes(5)

名前

syncinit – set serial line interface operating parameters

形式

```
/usr/sbin/syncinit device
     [ [baud_rate] | [keyword=value,]... | [single-word option]]
```

機能説明

The `syncinit` utility allows the user to modify some of the hardware operating modes common to synchronous serial lines. This can be useful in troubleshooting a link, or necessary to the operation of a communications package.

If run without options, `syncinit` reports the options as presently set on the port. If options are specified, the new settings are reported after they have been made.

オプション

Options to `syncinit` normally take the form of a keyword, followed by an equal sign and a value. The exception is that a baud rate may be specified as a decimal integer by itself. Keywords must begin with the value shown in the options table, but may contain additional letters up to the equal sign. For example, `loop=` and `loopback=` are equivalent.

The following options are supported:

Keyword	Value	Effect
`loop`	yes	Set the port to operate in `internal loopback` mode. The receiver is electrically disconnected from the DCE receive data input and tied to the outgoing transmit data line. Transmit data is available to the DCE. The Digital Phase-Locked Loop (DPLL) may not be used as a clock source in this mode. If no other clocking options have been specified, perform the equivalent of `txc=baud` and `rxc=baud`.
	no	Disable internal loopback mode. If no other clocking options have been specified, perform the equivalent of `txc=txc` and `rxc=rxc`.
`echo`	yes	Set the port to operate in `auto-echo` mode. The transmit data output is electrically disconnected from the transmitter and tied to the receive data input. Incoming receive data is still visible. Use of this mode in combination with local loopback mode has no value, and should be rejected by the device driver. The `auto-echo` mode is useful to make a system become the endpoint of a remote loopback test.
	no	Disable `auto-echo` mode.
`nrzi`	yes	Set the port to operate with `NRZI` data encoding.
	no	Set the port to operate with `NRZ` data encoding.
`txc`	`txc`	Transmit clock source will be the `TxC` signal (pin 15).
	`rxc`	Transmit clock source will be the `RxC` signal (pin 17).
	`baud`	Transmit clock source will be the internal `baud rate generator`.
	`pll`	Transmit clock source will be the output of the `DPLL` circuit.

rxc	rxc	Receive clock source will be the RxC signal (pin 17).
	txc	Receive clock source will be the TxC signal (pin 15).
	baud	Receive clock source will be the internal baud rate generator.
	pll	Receive clock source will be the output of the DPLL circuit.
speed	*integer*	Set the baud rate to *integer* bits per second.

There are also several single-word options that set one or more parameters at a time:

Keyword	Equivalent to Options:
external	txc=txc rxc=rxc loop=no
sender	txc=baud rxc=rxc loop=no
internal	txc=pll rxc=pll loop=no
stop	speed=0

使用例

例 1 Using syncinit

The following command sets the first CPU port to loop internally, using internal clocking and operating at 38400 baud:

```
example# syncinit zsh0 38400 loop=yes
device: /dev/zsh  ppa: 0
speed=38400, loopback=yes, echo=no, nrzi=no, txc=baud, rxc=baud
```

The following command sets the same port's clocking, local loopback and baud rate settings to their default values:

```
example# syncinit zsh0 stop loop=no
device: /dev/zsh  ppa: 0
speed=0, loopback=no, echo=no, nrzi=no, txc=txc, rxc=rxc
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

syncloop(1M), syncstat(1M), Intro(2), ioctl(2), attributes(5), zsh(7D)

診断

device missing minor device number	The name *device* does not end in a decimal number that can be used as a minor device number.

`bad speed:` *arg*	The string *arg* that accompanied the `speed=` option could not be interpreted as a decimal integer.
`Bad arg:` *arg*	The string *arg* did not make sense as an option.
`ioctl failure code =` *errno*	An `ioctl(2)` system called failed. The meaning of the value of *errno* may be found in `Intro(2)`.

警告

Do not use `syncinit` on an active serial link, unless needed to resolve an error condition. Do not use this command casually or without being aware of the consequences.

名前

syncloop – synchronous serial loopback test program

形式

`/usr/sbin/syncloop [-cdlstv] device`

機能説明

The `syncloop` command performs several loopback tests that are useful in exercising the various components of a serial communications link.

Before running a test, `syncloop` opens the designated port and configures it according to command line options and the specified test type. It announces the names of the devices being used to control the hardware channel, the channel number (ppa) corresponding to the *device* argument, and the parameters it has set for that channel. It then runs the loopback test in three phases.

The first phase is to listen on the port for any activity. If no activity is seen for at least four seconds, `syncloop` proceeds to the next phase. Otherwise, the user is informed that the line is active and that the test cannot proceed, and the program exits.

In the second phase, called the "first-packet" phase, `syncloop` attempts to send and receive one packet. The program will wait for up to four seconds for the returned packet. If no packets are seen after five attempts, the test fails with an excoriating message. If a packet is returned, the result is compared with the original. If the length and content do not match exactly, the test fails.

The final phase, known as the "multiple-packet" phase, attempts to send many packets through the loop. Because the program has verified the integrity of the link in the first-packet phase, the test will not fail after a particular number of timeouts. If a packet is not seen after four seconds, a message is displayed. Otherwise, a count of the number of packets received is updated on the display once per second. If it becomes obvious that the test is not receiving packets during this phase, the user may wish to stop the program manually. The number and size of the packets sent during this phase is determined by default values, or by command line options. Each returned packet is compared with its original for length and content. If a mismatch is detected, the test fails. The test completes when the required number of packets have been sent, regardless of errors.

After the multiple-packet phase has completed, the program displays a summary of the hardware event statistics for the channel that was tested. The display takes the following form:

```
CRC errors   Aborts   Overruns   Underruns   In<-Drops-> Out
        0         0          0           0   0             0
```

This is followed by an estimated line speed, which is an approximation of the bit rate of the line, based on the number of bytes sent and the actual time that it took to send them.

オプション

The options for `syncloop` are described in the following table:

Option	Parameter	Default	Description
-c	*packet_count*	100	Specifies the number of packets to be sent in the multiple-packet phase.
-d	*hex_data_byte*	*random*	Specifies that each packet will be filled with bytes with the value of *hex_data_byte*.
-l	*packet_length*	100	Specifies the length of each packet in bytes.
-s	*line_speed*	9600	Bit rate in bits per second.
-v			Sets verbose mode. If data errors occur, the expected and received data is displayed.
-t	*test_type*	*none*	A number, from 1 to 4, that specifies which test to perform. The values for *test_type* are as follows: 1: Internal loopback test. Port loopback is on. Transmit and receive clock sources are internal (baud rate generator). 2: External loopback test. Port loopback is off. Transmit and receive clock sources are internal. Requires a loopback plug suitable to the port under test. 3: External loopback test. Port loopback is off. Transmit and receive clock sources are external (modem). Requires that one of the local modem, the remote modem, or the remote system be set in a loopback configuration. 4: Test using predefined parameters. User defines hardware configuration and may select port parameters using the syncinit(1M) command.

All numeric options except -d are entered as decimal numbers (for example, -s 19200). If you do not provide the -t *test_type* option, syncloop prompts for it.

使用例

例 1 A sample display of using the syncloop command.

In the following command syncloop uses a packet length of 512 bytes over the first CPU port:

```
example# syncloop -l 512 zsh0
```

In response to the above command, syncloop prompts you for the test option you want.

The following command performs an internal loopback test on the first CPU port, using 5000 packets and a bit rate of 56Kbps:

```
example# syncloop -t 1 -s 56000 -c 5000 zsh0
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

syncinit(1M), syncstat(1M), attributes(5), zsh(7D)

診断

device `missing minor device number`	The name *device* does not end in a decimal number that can be used as a minor device number.
`invalid packet length:` *nnn*	The packet length was specified to be less than zero or greater than 4096.
`poll: nothing to read`	
`poll: nothing to read or write.`	The `poll(2)` system call indicates that there is no input pending and/or that output would be blocked if attempted.
`len` *xxx* `should be` *yyy*	The packet that was sent had a length of *yyy*, but was received with a length of *xxx*.
nnn `packets lost in outbound queueing`	
nnn `packets lost in inbound queueing`	A discrepancy has been found between the number of packets sent by `syncloop` and the number of packets the driver counted as transmitted, or between the number counted as received and the number read by the program.

警告

To allow its tests to run properly, as well as prevent disturbance of normal operations, `syncloop` should only be run on a port that is not being used for any other purpose at that time.

名前

syncstat – report driver statistics from a synchronous serial link

形式

`/usr/sbin/syncstat [-c] device [interval]`

機能説明

The `syncstat` command reports the event statistics maintained by a synchronous serial device driver. The report may be a single snapshot of the accumulated totals, or a series of samples showing incremental changes. Prior to these it prints the device name being used to query a particular device driver, along with a number indicating the channel number (ppa) under control of that driver.

Event statistics are maintained by a driver for each physical channel that it supports. They are initialized to zero at the time the driver module is loaded into the system, which may be either at boot time or when one of the driver's entry points is first called.

The *device* argument is the name of the serial device as it appears in the `/dev` directory. For example, `zsh0` specifies the first on-board serial device.

The following is a breakdown of `syncstat` output:

`speed`	The line speed the device has been set to operate at. It is the user's responsibility to make this value correspond to the modem clocking speed when clocking is provided by the modem.
`ipkts`	The total number of input packets.
`opkts`	The total number of output packets.
`undrun`	The number of transmitter underrun errors.
`ovrrun`	The number of receiver overrun errors.
`abort`	The number of aborted received frames.
`crc`	The number of received frames with CRC errors.
`isize`	The average size (in bytes) of input packets.
`osize`	The average size (in bytes) of output packets.

オプション

`-c` Clear the accumulated statistics for the device specified. This may be useful when it is not desirable to unload a particular driver, or when the driver is not capable of being unloaded.

interval `syncstat` samples the statistics every *interval* seconds and reports incremental changes. The output reports line utilization for input and output in place of average packet sizes. These are the relationships between bytes transferred and the baud rate, expressed as percentages. The loop repeats indefinitely, with a column heading printed every twenty lines for convenience.

使用例

例 1 Sample output from the syncstat command:

```
example# syncstat zsh0


speed ipkts opkts undrun ovrrun abort crc isize osize
9600  15716 17121   0      0      1    3   98    89
```

```
example# syncstat -c zsh0

speed ipkts opkts undrun ovrrun abort crc isize osize
9600   0     0     0      0      0     0    0     0
```

In the following sample output a new line of output is generated every five seconds:

```
example# syncstat zsh0 5

ipkts opkts undrun ovrrun abort crc iutil outil
12    10      0     0      0     0   5%    4%
22    60      0     0      0     0   3%    90%
36    14      0     0      0     1   51%   2%
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

syncinit(1M), syncloop(1M), attributes(5), zsh(7D)

診断

`bad interval:` *arg*	The argument *arg* is expected to be an interval and could not be understood.
device `missing minor device number`	The name *device* does not end in a decimal number that can be used as a minor device number.
`baud rate not set`	The *interval* option is being used and the baud rate on the device is zero. This would cause a divide-by-zero error when computing the line utilization statistics.

警告

Underrun, overrun, frame-abort, and CRC errors have a variety of causes. Communication protocols are typically able to handle such errors and initiate recovery of the transmission in which the error occurred. Small numbers of such errors are not a significant problem for most protocols. However, because the overhead involved in recovering from a link error can be much greater than that of normal operation, high error rates can greatly degrade overall link throughput. High error rates are often caused by problems in the link hardware, such as

cables, connectors, interface electronics or telephone lines. They may also be related to excessive load on the link or the supporting system.

The percentages for input and output line utilization reported when using the *interval* option may occasionally be reported as slightly greater than 100% because of inexact sampling times and differences in the accuracy between the system clock and the modem clock. If the percentage of use greatly exceeds 100%, or never exceeds 50%, then the baud rate set for the device probably does not reflect the speed of the modem.

名前

sysconfig – unconfigure or reconfigure a Solaris instance

形式

```
/usr/sbin/sysconfig

/usr/sbin/sysconfig configure [-s] [-c config_profile.xml | dir]
     [--destructive] [-g system]

/usr/sbin/sysconfig unconfigure [-s] [--destructive]
     [-g system]

/usr/sbin/sysconfig create-profile [-o output_file [-l logfile]
     [-v verbosity] [-b] [-g system]
```

機能説明

The `sysconfig` utility is the interface for unconfiguring and reconfiguring a Solaris instance. A Solaris instance is defined as a boot environment in either a global or a non-global zone.

There are three operations that are performed using the `sysconfig` utility: unconfiguration, configuration, and profile creation.

When `sysconfig` is called with the `unconfigure` subcommand, the system is unconfigured and left in an unconfigured state.

System configuration can occur either interactively or non-interactively. If the `configure` sub-command is invoked without a profile, the SCI Tool is activated and walks the user through the system configuration process. If the `configure` subcommand is invoked with a profile, then the configuration reads the profile and the configuration occurs non-interactively. The result in either case is a new system configuration.

The `sysconfig` command can also be used to generate a system configuration profile using the `create-profile` subcommand. The resulting profile is used with the `sysconfig configure` subcommand to configure systems non-interactively. Valid profile names include an `.xml` extension.

Configuration of a system can be performed either interactively, using the System Configuration Interactive (SCI) Tool, or non-interactively, using a system configuration profile.

The SCI tool configures the target system in an interactive way using a text user interface. It can also be used to collect information generated by the user that describes the desired configuration of the target system. The tool then generates a system configuration profile containing the desired system configuration.

The SCI tool supports configuration of freshly installed or unconfigured systems. It is designed to provide system configuration for newly created non-global zones and during text installation. If there is a need to modify the configuration of an already configured system utilizing SCI tool, such a system has to be unconfigured first before SCI tool can run.

The functional groupings that can be configured on a system are `network`, `location`, `users`, `identity`, and `kbd_layout`. Groupings can also be `unconfigured` and left in an `unconfigured` state. The default values for `unconfigured` groupings are shown below.

The following groupings are configurable.

```
Grouping        | Components            | Unconfigured State
-----------------------------------------------------------
identity        | system nodename       | unknown
-----------------------------------------------------------
kbd_layout      | Keyboard              | U.S. English
-----------------------------------------------------------
network         | network               | No network
-----------------------------------------------------------
location        | timezone              | UTC
                | locale                | C locale
-----------------------------------------------------------
users           | root                  | Empty root password
                | initial user account  | Remove user account
-----------------------------------------------------------
naming_services | DNS, NIS and LDAP     | No network naming
                | clients, nsswitch     | services
-----------------------------------------------------------
```

サブコマンド

This section describes supported subcommands and their associated options.

`unconfigure` [`-s`] [`-g` *system*] [`--destructive`]

Unconfigure a system and leave it in the `unconfigured` state.

`-s`

Shut the system down after the unconfiguration completes.

`-g` *system*

If `-g` is not specified, the user will be queried for confirmation before *system* configuration occurs.

`--destructive`

Do not preserve system data that is normally preserved during unconfiguration. By specifying this flag, the user indicates to any groupings unconfigured that data they would ordinarily preserve might be deleted.

`configure` [`-s`] [`-g` *system*] [`-c` *config_profile*.xml | *dir*] [`--destructive`]

Configure or reconfigure a system. The configure subcommand has access to the same options as the `unconfigure` subcommand. It also includes the following additional option.

`-c` *config_profile*.xml | *dir*

Provides a profile or a directory of profiles to apply during configuration. If a profile is applied, the configuration step occurs non-interactively. If no profile is provided, the interactive system configuration tool is used for the configuration of the system.

All profiles must have an `.xml` file extension.

If you supply a directory to `-c`, all profiles in that directory must be valid (correctly formed) configuration profiles.

`create-profile` [`-o` *output_file* [`-l` *logfile*] [`-v` *verbosity*] [`-b`] [`-g` *grouping*,...]
Run the SCI tool and create a system configuration profile. The default location for the profile is `/system/volatile/profile/sc_profile.xml`. The configuration generated is not applied to the system.

`-o` *output_file*
Replace the default profile location with *output_file* for the configuration profile.

`-l`
Location of the log file. Used only with `-o` option.

`-v`
Verbosity level. Used only with `-o` option.

`-b`
Black and white version of SCI tool. Used only with `-o` option.

使用例

例 1 Unconfiguring and Shutting Down

The following command unconfigures the system and leaves it in an `unconfigured` state. By default, if no grouping is specified, the groupings for the whole system are unconfigured.

```
# sysconfig unconfigure -s
```

例 2 Unconfiguring the System

The following command unconfigures the system groupings and leaves the system `unconfigured`.

```
# sysconfig unconfigure -g system
```

例 3 Reconfiguring System Using SCI Tool

The following command brings up the SCI Tool to reconfigure a system.

```
# sysconfig configure
```

例 4 Reconfiguring Using a Profile

The following command reconfigures a system using a profile.

```
# sysconfig configure -c some_profile.xml
```

例 5 Creating and Using a Profile

The following sequence of commands creates a profile, then uses it to reconfigure a system.

```
# sysconfig create-profile -o /tmp/myprofile.xml
# sysconfig configure -g system -c /tmp/myprofile.xml
```

例 6 Configuring the System in a Zone

The following command configures the system in a zone.

```
# zlogin ZONENAME
root@ZONENAME# sysconfig configure -g system
```

終了ステータス

0

Success.

>0

Failure.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
	system/install
	system/install/configuration
	system/library/install
Interface Stability	Committed

関連項目

svcprop(1), svcadm(1M), svccfg(1M), attributes(5), attributes(5)

名前

sysdef – output system definition

形式

```
/usr/sbin/sysdef [-i] [-n namelist]
/usr/sbin/sysdef [-h] [-d] [-i] [-D]
```

機能説明

The `sysdef` utility outputs the current system definition in tabular form. It lists all hardware devices, as well as pseudo devices, system devices, loadable modules, and the values of selected kernel tunable parameters.

It generates the output by analyzing the named bootable operating system file (*namelist*) and extracting the configuration information from it.

The default system *namelist* is `/dev/kmem`.

オプション

`-i`	Prints the configuration information from `/dev/kmem`. This is the default and only needs to be specified if the configuration information from both `/dev/kmem` and the system file specified with the "`-n` *namelist*" option is needed.
`-n` *namelist*	Specifies a *namelist* other than the default (`/dev/kmem`). The *namelist* specified must be a valid bootable operating system.
`-h`	Prints the identifier of the current host in hexadecimal. If `sysdef -h` is executed within a non-global zone and the zone emulates a host identifier, then the zone's host identifier is printed. This numeric value is not guaranteed to be unique.
`-d`	The output includes the configuration of system peripherals formatted as a device tree.
`-D`	For each system peripheral in the device tree, display the name of the device driver used to manage the peripheral.

使用例

例 1　Sample output format

The following example displays the format of the `sysdef -d` output:

```
example% sysdef -d
Node 'SUNW,Ultra-5_10', unit #-1
    Node 'packages', unit #-1 (no driver)
        Node 'terminal-emulator', unit #-1 (no driver)
        Node 'deblocker', unit #-1 (no driver)
        Node 'obp-tftp', unit #-1 (no driver)
        Node 'disk-label', unit #-1 (no driver)
        Node 'SUNW,builtin-drivers', unit #-1 (no driver)
        Node 'sun-keyboard', unit #-1 (no driver)
        Node 'ufs-file-system', unit #-1 (no driver)
    Node 'chosen', unit #-1 (no driver)
    Node 'openprom', unit #-1 (no driver)
        Node 'client-services', unit #-1 (no driver)
```

例 1 Sample output format (続き)

```
Node 'options', unit #0
Node 'aliases', unit #-1 (no driver)
Node 'memory', unit #-1 (no driver)
Node 'virtual-memory', unit #-1 (no driver)
Node 'pci', unit #0
    Node 'pci', unit #0
        Node 'ebus', unit #0
            Node 'auxio', unit #-1 (no driver)
            Node 'power', unit #0
            Node 'SUNW,pll', unit #-1 (no driver)
            Node 'se', unit #0 (no driver)
            Node 'su', unit #0
            Node 'su', unit #1
            Node 'ecpp', unit #-1 (no driver)
            Node 'fdthree', unit #0
            Node 'eeprom', unit #-1 (no driver)
            Node 'flashprom', unit #-1 (no driver)
            Node 'SUNW,CS4231', unit #0 (no driver)
        Node 'network', unit #0
        Node 'SUNW,m64B', unit #0
        Node 'ide', unit #0
            Node 'disk', unit #-1 (no driver)
            Node 'cdrom', unit #-1 (no driver)
            Node 'sd', unit #1
            Node 'dad', unit #1
    Node 'pci', unit #-1 (no driver)
Node 'SUNW,UltraSPARC-IIi', unit #-1 (no driver)
Node 'pseudo', unit #0
```

[output truncated]

ファイル

`/dev/kmem` default operating system image

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`hostid(1)`, `prtconf(1M)`, `nlist(3ELF)`, `attributes(5)`, `zones(5)`

名前

syseventadm – sysevent event specification administration

形式

```
syseventadm add [-R rootdir] [-v vendor] [-p publisher]
     [-c class] [-s subclass] [-u username] path [args]

syseventadm remove [-R rootdir] [-v vendor] [-p publisher]
     [-c class] [-s subclass] [-u username] [path [args]]

syseventadm list [-R rootdir] [-v vendor] [-p publisher]
     [-c class] [-s subclass] [-u username] [path [args]]

syseventadm restart
```

機能説明

The `syseventadm` command is an administrative front-end to add, remove and list `sysevent` event handlers. You can also restart the sysevent daemon by use of the restart command. `syseventadm` can only be run by root.

The `syseventadm add` command adds a handler for a `sysevent` event specified by at least one of vendor, publisher or class. If *class* is specified, it may be qualified with a *sub-class*. Only the values specified for *vendor*, *publisher*, *class* and *sub-class* when adding the handler are matched against `sysevent` events to determine if the specification matches the event and the handler should be run. *path* is the full pathname of the command to be run in response to matching events, with optional arguments (*args*). If username is specified, the command is invoked as user *username*, otherwise as root.

The `syseventadm remove` command removes handlers for matching sysevent event specifications. Event specifications may be matched by specifying at least one of *vendor*, *publisher*, *class*, *username* or *path*. If *class* is specified, it may be qualified with a *sub-class*. Any of *vendor*, *publisher*, *class*, *sub-class*, *username*, *path* or *args* not specified match the corresponding fields of all events. Handlers for all matching specifications are removed.

The `syseventadm list` command lists the handlers for matching `sysevent` event specifications using the same match criteria as the `remove` command but without the requirement that at least one of *vendor*, *publisher*, *class*, *username* or *path* be specified. With no match criteria, all specifications are listed. The list command output format is: [vendor=*vendor*] [publisher=*publisher*] [class=*class*] [subclass=*subclass*] [username=*username*] *path* [*args*] where each of *class*, *sub-class*, *vendor*, *publisher* and *username* is listed only if part of the match criteria for the listed specification.

The `syseventadm restart` command informs the `syseventd` daemon to reread the `sysevent` registry after a change has been made by adding or removing one or more `sysevent` handler specifications.

Argument Macro Substitution

The `sysevent` handling facility provides extensive macro capability for constructing the command line arguments to be executed in response to an event. Macro expansion applies only to the command line *args* specified for an event handler, with macros expanded with data from the event itself. Pre-defined macros are provided for the event *class*, *subclass*, *publisher* and *vendor* information. Macros not matching one of the pre-defined macro names cause the

attribute list attached to the event to be searched for an attribute of that name, with the value of the matching attribute substituted on the command line.

Macros are introduced by the `$` character, with the macro name being the following token separated by a SPACE or TAB character. If the macro name is embedded in text, it may be delineated by `${` and `}`. A `\` before the `$` causes macro expansion not to occur.

$*class*	The class string defining the event
$*publisher*	The publisher string defining the event
$*sequence*	The sequence number of the event.
$*subclass*	The subclass string defining the event
$*timestamp*	The timestamp of the event.
$*vendor*	The vendor string defining the event

Macro names other than those pre-defined are compared against the attribute list provided with the event. An attribute with name matching the macro name causes the value of the attribute to be substituted as ASCII text on the generated command line.

Use of a macro for which no attribute with that name is defined, or for which multiple attributes with that name are provided, cause an error and the command is not invoked.

Attributes with signed data types (`DATA_TYPE_INT16`, `DATA_TYPE_INT32` and `DATA_TYPE_INT64`) are expanded as decimal digits.

Attributes with unsigned data types (`DATA_TYPE_BYTE`, `DATA_TYPE_UINT16`, `DATA_TYPE_UINT32`, `DATA_TYPE_UINT64` and `DATA_TYPE_HTTIME`) are expanded as hexadecimal digits with a `0x` prefix.

Attributes with string data type (`DATA_TYPE_STRING`) are expanded with the string data. The data is not quoted. If if it desired that the quoted strings be generated on the command line, put quotes around the macro call in the arguments.

Array types are expanded with each element expanded as defined for that scalar type, with a space separating each element substitution.

オプション

The `add`, `list` and `remove` subcommands support the following options:

`-c` *class*	Specify the event class, *class*.
`-p` *publisher*	Specify the event publisher, *publisher*.
`-R` *rootdir*	Specify an alternate root path, *rootdir*.

注 – The root file system of any non-global zones must not be referenced with the `-R` option. Doing so might damage the global zone's file system, might

compromise the security of the global zone, and might damage the non-global zone's file system. See zones(5).

-s *subclass* Specify the event subclass, *subclass*.

-u *username* Specify the username (*username*) to invoke the command.

-v *vendor* Specify the vendor (*vendor*) that defines the event. Events defined by third-party software should specify the company's stock symbol as *vendor*. Sun-defined events use SUNW.

オペランド

The add, list and remove subcommands support the following options:

args Command arguments

path Full path of command to be run in response to event

使用例

例 1 Adding an Event Handler

The following example adds an event handler for an event defined by vendor MYCO ("My Company"), class EC_ENV and sub-class ESC_ENV_TEMP. The command to be run is /opt/MYCOenv/bin/ec_env_temp, with arguments being the class name, sub-class name and pathname derived from the event attributes. The $ characters are preceded by a backslash to circumvent shell interpretation. There is no need to restart the service after the change since the registry is maintained on $ALTROOT.

```
# syseventadm add -R \$ALTROOT -v MYCO -c EC_ENV -s ESC_ENV_TEMP \
     /opt/MYCOenv/bin/ec_env_temp \$class \$subclass \$pathname
```

Note the caveat on the use of the -R option in the description of that option, above.

例 2 Removing an Event Handler

The following example removes the event handler added in Example 1.

```
# syseventadm remove -R \$ALTROOT -v MYCO -c EC_ENV -s ESC_ENV_TEMP \
     /opt/MYCOenv/bin/ec_env_temp \${class} \${subclass} \${pathname}
```

Note the caveat on the use of the -R option in the description of that option, above.

例 3 Listing Event Handlers

The following example lists all event handlers for events of class EC_ENV, subclass ESC_ENV_TEMP, as defined by vendor MYCO:

```
# syseventadm list -v MYCO -c EC_ENV -s ESC_ENV_TEMP \
     vendor=MYCO class=EC_ENV subclass=ESC_ENV_TEMP \
     /opt/MYCOenv/bin/ec_env_temp \${class} \${subclass} \${pathname}
```

例 4 Listing Event Handlers

The following example lists all event handlers defined by vendor VRTS.

```
# syseventadm list -v VRTS
```

例 5 Removing Event Handlers

The following example removes all event handlers defined by vendor VRTS, and restarts service.

```
# syseventadm remove -v VRTS
# syseventadm restart
```

例 6 Listing All Event Handlers Specified to Run a Command

The following example lists all event handlers specified to run the command /opt/MYCOenv/bin/ec_env_temp:

```
# syseventadm list /opt/MYCOenv/bin/ec_env_temp
```

例 7 Removing Event Handlers and Restarting Service

The following example removes all event handlers specified to run the command /opt/MYCOenv/bin/ec_env_temp, and restarts service:

```
# syseventadm remove /opt/MYCOenv/bin/ec_env_temp
# syseventadm restart
```

終了ステータス

The following exit values are returned:

0 Successful completion.

1 No matching event specification found (remove or list commands only).

2 Incorrect command usage.

3 Permission denied.

4 Command failed.

5 Out of memory.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

sysevent d(1M), sysevent_post_event(3SYSEVENT), attributes(5), ddi_log_sysevent(9F)

注意事項

To avoid upgrade problems, packages delivering a sysevent event handler should install the event handler by running `syseventadm` from the package's postinstall script. The event handler can then be removed by running `syseventadm` from the package's preremove script using the same arguments as when added.

名前

syseventconfd – kernel system event command invocation daemon

形式

`/usr/lib/sysevent/syseventconfd [-r rootdir]`

機能説明

`syseventconfd` is the user-level daemon that invokes user-level commands in response to kernel system events received from `sysevent d(1M)`.

オプション

The following options are supported:

`-r` *rootdir* Cause `syseventconfd` to use an alternate root path when creating its door. The root path must match the root path used to invoke `sysevent d`.

ファイル

`/etc/sysevent/syseventconfd_event_service`
`syseventconfd` event service door file

`/usr/lib/sysevent/modules/sysevent_conf_mod.so`
`sysevent d` loadable module (SLM) managing `sysevent.conf` files

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`svcs(1)`, `svcadm(1M)`, `sysevent d(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `syseventconfd` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/sysevent:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

syseventd – kernel system event notification daemon

形式

`/usr/lib/sysevent/syseventd [-d debug_level] [-r rootdir]`

機能説明

`syseventd` is a user-level daemon that accepts delivery of system event buffers from the kernel. Once an event buffer has been delivered to `syseventd`, it, in turn, attempts to propagate the event to all interested end event subscribers.

Event subscribers take the form of a `syseventd` loadable module (SLM). sysevent passes the event buffer to each of its subscribers and in return expects a notification as to the successful or unsuccessful delivery attempt.

Upon successful delivery of the event buffer to all interested event subscribers, `syseventd` frees the event buffer from the kernel event queue.

オプション

The following option is supported:

`-d` *debug_level* Enable debug mode. Messages are printed to the invoking user's terminal.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`non-zero` An error occurred.

ファイル

`/etc/sysevent/syseventd_daemon.lock`
daemon lock file

`/etc/sysevent/sysevent_door`
kernel to `syseventd` door file

`/usr/lib/sysevent/modules`
SLM directory repository

`/usr/platform/‘uname –i‘/lib/sysevent/modules`
SLM directory repository

`/usr/platform/‘uname –m‘/lib/sysevent/modules`
SLM directory repository

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`svcs(1)`, `svcadm(1M)`, `syseventconfd(1M)`, `attributes(5)`, `smf(5)`

注意事項

The `sysevent d` service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/sysevent:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

syslogd – log system messages

形式

```
/usr/sbin/syslogd [-d] [-f configfile] [-m markinterval]
     [-p path] [-t | -T]
```

機能説明

syslogd reads and forwards system messages to the appropriate log files or users, depending upon the priority of a message and the system facility from which it originates. The configuration file /etc/syslog.conf (see syslog.conf(4)) controls where messages are forwarded. syslogd logs a mark (timestamp) message every *markinterval* minutes (default 20) at priority LOG_INFO to the facility whose name is given as mark in the syslog.conf file.

A system message consists of a single line of text, which may be prefixed with a priority code number enclosed in angle-brackets (< >); priorities are defined in <sys/syslog.h>.

syslogd reads from the STREAMS log driver, /dev/log, and from any transport provider specified in /etc/netconfig, /etc/net/transport/hosts, and /etc/net/transport/services.

syslogd reads the configuration file when it starts up, and again whenever it receives a HUP signal (see signal.h(3HEAD), at which time it also closes all files it has open, re-reads its configuration file, and then opens only the log files that are listed in that file. syslogd exits when it receives a TERM signal.

As it starts up, syslogd creates the file /var/run/syslog.pid, if possible, containing its process identifier (PID).

If message ID generation is enabled (see log(7D)), each message will be preceded by an identifier in the following format: [ID *msgid facility*.*priority*]. *msgid* is the message's numeric identifier described in msgid(1M). *facility* and *priority* are described in syslog.conf(4). [ID 123456 kern.notice] is an example of an identifier when message ID generation is enabled.

If the message originated in a loadable kernel module or driver, the kernel module's name (for example, ufs) will be displayed instead of unix. See EXAMPLES for sample output from syslogd with and without message ID generation enabled.

In an effort to reduce visual clutter, message IDs are not displayed when writing to the console; message IDs are only written to the log file. See 使用例.

The /etc/default/syslogd file contains the following default parameter settings, which are in effect if neither the -t nor -T option is selected. See FILES.

The recommended way to allow or disallow message logging is through the use of the service management facility (smf(5)) property:

```
svc:/system/system-log/config/log_from_remote
```

This property specifies whether remote messages are logged. log_from_remote=true is equivalent to the -t command-line option and false is equivalent to the -T command-line option. The default value for -log_from_remote is false. See NOTES, below.

`LOG_FROM_REMOTE`
: Specifies whether remote messages are logged. `LOG_FROM_REMOTE=NO` is equivalent to the `-t` command-line option. The default value for `LOG_FROM_REMOTE` is `YES`.

オプション

The following options are supported:

`-d`
: Turn on debugging. This option should only be used interactively in a root shell once the system is in multi-user mode. It should *not* be used in the system start-up scripts, as this will cause the system to hang at the point where `syslogd` is started.

`-f` *configfile*
: Specify an alternate configuration file.

`-m` *markinterval*
: Specify an interval, in minutes, between mark messages.

`-p` *path*
: Specify an alternative log device name. The default is `/dev/log`.

`-T`
: Enable the `syslogd` UDP port to turn on logging of remote messages. This is the default behavior. See ファイル.

`-t`
: Disable the `syslogd` UDP port to turn off logging of remote messages. See ファイル.

使用例

例 1 syslogd Output Without Message ID Generation Enabled

The following example shows the output from `syslogd` when message ID generation *is not* enabled:

```
Sep 29 21:41:18 cathy unix: alloc /: file system full
```

例 2 syslogd Output with ID generation Enabled

The following example shows the output from `syslogd` when message ID generation *is* enabled. The message ID is displayed when writing to log file`/var/adm/messages`.

```
Sep 29 21:41:18 cathy ufs: [ID 845546 kern.notice]
                                    alloc /: file system full
```

例 3 syslogd Output with ID Generation Enabled

The following example shows the output from `syslogd` when message ID generation *is* enabled when writing to the console. Even though message ID is enabled, the message ID is not displayed at the console.

```
Sep 29 21:41:18 cathy ufs: alloc /: file system full
```

例 4 Enabling Acceptance of UDP Messages from Remote Systems

The following commands enable `syslogd` to accept entries from remote systems.

```
# svccfg -s svc:/system/system-log setprop config/log_from_remote = true
# svcadm restart svc:/system/system-log
```

ファイル

`/etc/syslog.conf`
Configuration file

`/var/run/syslog.pid`
Process ID

`/etc/default/syslogd`
Contains default settings. You can override some of the settings by command-line options.

`/dev/log`
STREAMS log driver

`/etc/netconfig`
Transport providers available on the system

`/etc/net/transport/hosts`
Network hosts for each transport

`/etc/net/transport/services`
Network services for each transport

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`logger(1)`, `svcs(1)`, `msgid(1M)`, `svcadm(1M)`, `svccfg(1M)`, `syslog(3C)`, `syslog.conf(4)`, `attributes(5)`, `signal.h(3HEAD)`, `smf(5)`, `log(7D)`

注意事項

The `mark` message is a system time stamp, and so it is only defined for the system on which `syslogd` is running. It can not be forwarded to other systems.

When `syslogd` receives a `HUP` signal, it attempts to complete outputting pending messages, and close all log files to which it is currently logging messages. If, for some reason, one (or more) of these files does not close within a generous grace period, `syslogd` discards the pending messages, forcibly closes these files, and starts reconfiguration. If this shutdown procedure is disturbed by an unexpected error and `syslogd` cannot complete reconfiguration, `syslogd` sends a mail message to the superuser on the current system stating that it has shut down, and exits.

Care should be taken to ensure that each window displaying messages forwarded by `syslogd` (especially console windows) is run in the system default locale (which is `syslogd`'s locale). If

this advice is not followed, it is possible for a syslog message to alter the terminal settings for that window, possibly even allowing remote execution of arbitrary commands from that window.

The syslogd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/system-log:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

When syslogd is started by means of svcadm(1M), if a value is specified for LOG_FROM_REMOTE in the /etc/defaults/syslogd file, the SMF property svc:/system/system-log/config/log_from_remote is set to correspond to the LOG_FROM_REMOTE value and the /etc/default/syslogd file is modified to replace the LOG_FROM_REMOTE specification with the following comment:

```
# LOG_FROM_REMOTE is now set using svccfg(1m), see syslogd(1m).
```

If neither LOG_FROM_REMOTE nor svc:/system/system-log/config/log_from_remote are defined, the default is to log remote messages.

On installation, the initial value of svc:/system/system-log/config/log_from_remote is false.

名前

tapes – creates /dev entries for tape drives attached to the system

形式

`/usr/sbin/tapes [-r root_dir]`

機能説明

devfsadm(1M) is now the preferred command for `/dev` and `/devices` and should be used instead of `tapes`.

`tapes` creates symbolic links in the `/dev/rmt` directory to the actual tape device special files under the `/devices` directory tree. `tapes` searches the kernel device tree to see what tape devices are attached to the system. For each equipped tape drive, the following steps are performed:

1. The `/dev/rmt` directory is searched for a `/dev/rmt/`*n* entry that is a symbolic link to the `/devices` special node of the current tape drive. If one is found, this determines the logical controller number of the tape drive.
2. The rest of the special devices associated with the drive are checked, and incorrect symbolic links are removed and necessary ones added.
3. If none are found, a new logical controller number is assigned (the lowest-unused number), and new symbolic links are created for all the special devices associated with the drive.

`tapes` does not remove links to non-existent devices; these must be removed by hand.

`tapes` is run each time a reconfiguration-boot is performed, or when add_drv(1M) is executed.

Notice to Driver Writers

tapes(1M) considers all devices with the node type `DDI_NT_TAPE` to be tape devices; these devices must have their minor name created with a specific format. The minor name encodes operational modes for the tape device and consists of an ASCII string of the form [`l`,`m`,`h`,`c`,`u`][`b`][`n`].

The first character set is used to specify the tape density of the device, and are named low (`l`), medium (`m`), high (`h`), compressed (`c`), and ultra (`u`). These specifiers only express a relative density; it is up to the driver to assign specific meanings as needed. For example, 9 track tape devices interpret these as actual bits-per-inch densities, where `l` means 800 BPI, `m` means 1600 BPI , and `h` means 6250 BPI, whereas 4mm DAT tapes defines `l` as standard format, and `m`, `h`, `c` and `u` as compressed format. Drivers may choose to implement any or all of these format types.

During normal tape operation (non-BSD behavior), once an EOF mark has been reached, subsequent reads from the tape device return an error. An explicit IOCTL must be issued to space over the EOF mark before the next file can be read. `b` instructs the device to observe BSD behavior, where reading at EOF will cause the tape device to automatically space over the EOF mark and begin reading from the next file.

`n` or no-rewind-on-close instructs the driver to not rewind to the beginning of tape when the device is closed. Normal behavior for tape devices is to reposition to BOT when closing. See mtio(7I).

The minor number for tape devices should be created by encoding the device's instance number using the tape macro MTMINOR and ORing in the proper combination of density, BSD behavior, and no-rewind flags. See mtio(7I).

To prevent tapes from attempting to automatically generate links for a device, drivers must specify a private node type and refrain from using the node type string DDI_NT_TAPE when callingddi_create_minor_node(9F).

オプション

The following options are supported:

-r *root_dir* Causes tapes to presume that the /dev/rmt directory tree is found under *root_dir*, not directly under /.

エラー

If tapes finds entries of a particular logical controller linked to different physical controllers, it prints an error message and exits without making any changes to the /dev directory, since it cannot determine which of the two alternative logical to physical mappings is correct. The links should be manually corrected or removed before another reconfiguration boot is performed.

使用例

例 1 Creating Tape Device Nodes From Within the Driver's attach() Function

This example demonstrates creating tape device nodes from within the xktape driver's attach(9E) function.

```
#include <sys/mtio.h>
struct tape_minor_info {
    char *minor_name;
    int   minor_mode;
};
/*
 * create all combinations of logical tapes
*/
static struct tape_minor_info example_tape[] = {
   {"",    0},                     /* default tape */
   {"l",   MT_DENSITY1},
   {"lb",  MT_DENSITY1 | MT_BSD},
   {"lbn", MT_DENSITY1 | MT_BSD | MT_NOREWIND},
   {"m",   MT_DENSITY2},
   {"mb",  MT_DENSITY2 | MT_BSD},
   {"mbn", MT_DENSITY2 | MT_BSD | MT_NOREWIND},
   {"h",   MT_DENSITY3},
   {"hb",  MT_DENSITY3 | MT_BSD},
   {"hbn", MT_DENSITY3 | MT_BSD | MT_NOREWIND},
   {"c",   MT_DENSITY4},
   {"cb",  MT_DENSITY4 | MT_BSD},
   {"cbn", MT_DENSITY4| MT_BSD | MT_NOREWIND},
   {NULL,  0},
};
```

例 1　Creating Tape Device Nodes From Within the Driver's attach() Function　(続き)

```
int
xktapeattach(dev_info_t *dip, ddi_attach_cmd_t cmd)
{
   int instance;
   struct tape_minor_info *mdp;
      /* other stuff in attach... */
   instance = ddi_get_instance(dip);

   for (mdp = example_tape; mdp->minor_name != NULL; mdp++) {
            ddi_create_minor_node(dip, mdp->minor_name, S_IFCHR,
                 (MTMINOR(instance) | mdp->minor_mode), DDI_NT_TAPE, 0);
 }
```

Installing the `xktape` driver on a Sun Fire 4800, with the driver controlling a SCSI tape (target 4 attached to an isp(7D) SCSI HBA) and performing a reconfiguration-boot creates the following special files in `/devices`.

```
# ls -l /devices/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4
crw-rw-rw-   1 root sys   33,136 Aug 29 00:02  xktape@4,0:
crw-rw-rw-   1 root sys   33,200 Aug 29 00:02  xktape@4,0:b
crw-rw-rw-   1 root sys   33,204 Aug 29 00:02  xktape@4,0:bn
crw-rw-rw-   1 root sys   33,152 Aug 29 00:02  xktape@4,0:c
crw-rw-rw-   1 root sys   33,216 Aug 29 00:02  xktape@4,0:cb
crw-rw-rw-   1 root sys   33,220 Aug 29 00:02  xktape@4,0:cbn
crw-rw-rw-   1 root sys   33,156 Aug 29 00:02  xktape@4,0:cn
crw-rw-rw-   1 root sys   33,144 Aug 29 00:02  xktape@4,0:h
crw-rw-rw-   1 root sys   33,208 Aug 29 00:02  xktape@4,0:hb
crw-rw-rw-   1 root sys   33,212 Aug 29 00:02  xktape@4,0:hbn
crw-rw-rw-   1 root sys   33,148 Aug 29 00:02  xktape@4,0:hn
crw-rw-rw-   1 root sys   33,128 Aug 29 00:02  xktape@4,0:l
crw-rw-rw-   1 root sys   33,192 Aug 29 00:02  xktape@4,0:lb
crw-rw-rw-   1 root sys   33,196 Aug 29 00:02  xktape@4,0:lbn
crw-rw-rw-   1 root sys   33,132 Aug 29 00:02  xktape@4,0:ln
crw-rw-rw-   1 root sys   33,136 Aug 29 00:02  xktape@4,0:m
crw-rw-rw-   1 root sys   33,200 Aug 29 00:02  xktape@4,0:mb
crw-rw-rw-   1 root sys   33,204 Aug 29 00:02  xktape@4,0:mbn
crw-rw-rw-   1 root sys   33,140 Aug 29 00:02  xktape@4,0:mn
crw-rw-rw-   1 root sys   33,140 Aug 29 00:02  xktape@4,0:n
```

`/dev/rmt` will contain the logical tape devices (symbolic links to tape devices in `/devices`).

```
# ls -l /dev/rmt
/dev/rmt/0    -> ../../devices/[....]/xktape@4,0:
/dev/rmt/0b   -> ../../devices/[....]/xktape@4,0:b
/dev/rmt/0bn  -> ../../devices/[....]/xktape@4,0:bn
```

例 1 Creating Tape Device Nodes From Within the Driver's attach() Function *(続き)*

```
/dev/rmt/0c   -> ../../devices/[....]/xktape@4,0:c
/dev/rmt/0cb  -> ../../devices/[....]/xktape@4,0:cb
/dev/rmt/0cbn -> ../../devices/[....]/xktape@4,0:cbn
/dev/rmt/0cn  -> ../../devices/[....]/xktape@4,0:cn
/dev/rmt/0h   -> ../../devices/[....]/xktape@4,0:h
/dev/rmt/0hb  -> ../../devices/[....]/xktape@4,0:hb
/dev/rmt/0hbn -> ../../devices/[....]/xktape@4,0:hbn
/dev/rmt/0hn  -> ../../devices/[....]/xktape@4,0:hn
/dev/rmt/0l   -> ../../devices/[....]/xktape@4,0:l
/dev/rmt/0lb  -> ../../devices/[....]/xktape@4,0:lb
/dev/rmt/0lbn -> ../../devices/[....]/xktape@4,0:lbn
/dev/rmt/0ln  -> ../../devices/[....]/xktape@4,0:ln
/dev/rmt/0m   -> ../../devices/[....]/xktape@4,0:m
/dev/rmt/0mb  -> ../../devices/[....]/xktape@4,0:mb
/dev/rmt/0mbn -> ../../devices/[....]/xktape@4,0:mbn
/dev/rmt/0mn  -> ../../devices/[....]/xktape@4,0:mn
/dev/rmt/0n   -> ../../devices/[....]/xktape@4,0:n
```

ファイル

`/dev/rmt/*` logical tape devices

`/devices/*` tape device nodes

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`add_drv(1M)`, `devfsadm(1M)`, `attributes(5)`, `isp(7D)`, `devfs(7FS)`, `mtio(7I)`, `attach(9E)`, `ddi_create_minor_node(9F)`

『Writing Device Drivers』

使用上の留意点

`tapes` silently ignores malformed minor device names.

名前

tcxconfig – configure default linearity of 24-bit TrueColor Visual for OpenWindows

形式

```
/usr/sbin/tcxconfig [linear | nonlinear]
```

機能説明

The `tcxconfig` script changes the default linearity of a 24-bit TrueColor Visual for OpenWindows on a system with an S24 frame buffer. When the S24 graphics driver for OpenWindows is installed, the default 24-bit TrueColor Visual is nonlinear. You can run `tcxconfig` with an argument that specifies the setting you want.

OpenWindows should not be running when you execute the `tcxconfig` script with an option. Start OpenWindows after `tcxconfig` has set the linearity you desire.

オプション

If you specify no option, `tcxconfig` displays the current default setting.

You must become superuser before you can execute `tcxconfig` with one of the following options.

`linear` Set linear visual to be the default 24-bit TrueColor Visual. This means colors will be gamma-corrected.

`nonlinear` Set nonlinear visual to be the default 24-bit TrueColor Visual.

終了ステータス

The following exit values are returned:

`0` success

`1` an error has occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWtcxow

関連項目

`attributes(5)`

名前

th_define – create fault injection test harness error specifications

形式

```
th_define [-n name -i instance| -P path] [-a acc_types]
     [-r reg_number] [-l offset [length]]
     [-c count [failcount]] [-o operator [operand]]
     [-f acc_chk] [-w max_wait_period [report_interval]]

or

th_define [-n name -i instance| -P path]
     [-a log [acc_types] [-r reg_number] [-l offset [length]]]
     [-c count [failcount]] [-s collect_time] [-p policy]
     [-x flags] [-C comment_string]
     [-e fixup_script [args]]

or

th_define [-h]
```

機能説明

The `th_define` utility provides an interface to the `bus_ops` fault injection `bofi` device driver for defining error injection specifications (referred to as errdefs). An errdef corresponds to a specification of how to corrupt a device driver's accesses to its hardware. The command line arguments determine the precise nature of the fault to be injected. If the supplied arguments define a consistent errdef, the `th_define` process will store the errdef with the `bofi` driver and suspend itself until the criteria given by the errdef become satisfied (in practice, this will occur when the access counts go to zero).

You use the `th_manage(1M)` command with the `start` option to activate the resulting errdef. The effect of `th_manage` with the `start` option is that the `bofi` driver acts upon the errdef by matching the number of hardware accesses—specified in *count*, that are of the type specified in *acc_types*, made by instance number *instance*—of the driver whose name is *name*, (or by the driver instance specified by *path*) to the register set (or DMA handle) specified by *reg_number*, that lie within the range *offset* to *offset* + *length* from the beginning of the register set or DMA handle. It then applies *operator* and *operand* to the next *failcount* matching accesses.

If *acc_types* includes `log`, `th_define` runs in automatic test script generation mode, and a set of test scripts (written in the Korn shell) is created and placed in a sub-directory of the current directory with the name <*driver*>`.test.`<*id*> (for example, `glm.test.978177106`). A separate, executable script is generated for each access handle that matches the logging criteria. The log of accesses is placed at the top of each script as a record of the session. If the current directory is not writable, file output is written to standard output. The base name of each test file is the driver name, and the extension is a number that discriminates between different access handles. A control script (with the same name as the created test directory) is generated that will run all the test scripts sequentially.

Executing the scripts will install, and then activate, the resulting error definitions. Error definitions are activated sequentially and the driver instance under test is taken offline and brought back online before each test (refer to the `-e` option for more information). By default, logging applies to all `PIO` accesses, all interrupts, and all DMA accesses to and from areas

mapped for both reading and writing. You can constrain logging by specifying additional *acc_types*, *reg_number*, *offset* and *length*. Logging will continue for *count* matching accesses, with an optional time limit of *collect_time* seconds.

Either the `-n` or `-P` option must be provided. The other options are optional. If an option (other than `-a`) is specified multiple times, only the final value for the option is used. If an option is not specified, its associated value is set to an appropriate default, which will provide maximal error coverage as described below.

オプション

The following options are available:

`-n` *name*	Specify the name of the driver to test. (String)
`-i` *instance*	Test only the specified driver instance (-1 matches all instances of driver). (Numeric)
`-P` *path*	Specify the full device path of the driver to test. (String)
`-r` *reg_number*	Test only the given register set or DMA handle (-1 matches all register sets and DMA handles). (Numeric)
`-a` *acc_types*	Only the specified access types will be matched. Valid values for the *acc_types* argument are `log`, `pio`, `pio_r`, `pio_w`, `dma`, `dma_r`, `dma_w` and `intr`. Multiple access types, separated by spaces, can be specified. The default is to match all hardware accesses. If *acc_types* is set to `log`, logging will match all `PIO` accesses, interrupts and DMA accesses to and from areas mapped for both reading and writing. `log` can be combined with other *acc_types*, in which case the matching condition for logging will be restricted to the specified addional *acc_types*. Note that `dma_r` will match only DMA handles mapped for reading only; `dma_w` will match only DMA handles mapped for writing only; `dma` will match only DMA handles mapped for both reading and writing.
`-l` *offset* [*length*]	Constrain the range of qualifying accesses. The *offset* and *length* arguments indicate that any access of the type specified with the `-a` option, to the register set or DMA handle specified with the `-r` option, lie at least *offset* bytes into the register set or DMA handle and at most *offset* + *length*

bytes into it. The default for *offset* is 0. The default for *length* is the maximum value that can be placed in an `offset_t` C data type (see `types.h`). Negative values are converted into unsigned quantities. Thus, `th_define -l 0 -1` is maximal.

`-c` *count* [*failcount*]

Wait for *count* number of matching accesses, then apply an operator and operand (see the `-o` option) to the next *failcount* number of matching accesses. If the access type (see the `-a` option) includes logging, the number of logged accesses is given by *count* + *failcount* - 1. The -1 is required because the last access coincides with the first faulting access.

Note that access logging may be combined with error injection if *failcount* and *operator* are nonzero and if the access type includes logging and any of the other access types (`pio`, `dma` and `intr`) See the description of access types in the definition of the `-a` option, above.

When the *count* and *failcount* fields reach zero, the status of the errdef is reported to standard output. When all active errdefs created by the `th_define` process complete, the process exits. If *acc_types* includes `log`, *count* determines how many accesses to log. If *count* is not specified, a default value is used. If *failcount* is set in this mode, it will simply increase the number of accesses logged by a further *failcount* - 1.

`-o` *operator* [*operand*]

For qualifying PIO read and write accesses, the value read from or written to the hardware is corrupted according to the value of *operator*:

`EQ` *operand* is returned to the driver.

`OR` *operand* is bitwise ORed with the real value.

`AND` *operand* is bitwise ANDed with the real value.

`XOR` *operand* is bitwise XORed with the real value.

For PIO write accesses, the following operator is allowed:

`NO` Simply ignore the driver's attempt to write to the hardware.

Note that a driver performs PIO via the `ddi_get`*X*`()`, `ddi_put`*X*`()`, `ddi_rep_get`*X*`()` and `ddi_rep_put`*X*`()` routines (where *X* is 8, 16, 32 or 64). Accesses made using `ddi_get`*X*`()` and `ddi_put`*X*`()` are treated as a single access, whereas an access made using the `ddi_rep_*`(9F) routines are broken down into their respective number of accesses, as given by the *repcount* parameter to these DDI calls. If the access is performed via a DMA handle, *operator* and *value* are applied to every access that comprises the DMA request. If interference with interrupts has been requested then the operator may take any of the following values:

`DELAY` After *count* accesses (see the `-c` option), delay delivery of the next *failcount* number of interrupts for *operand* number of microseconds.

`LOSE` After *count* number of interrupts, fail to deliver the next *failcount* number of real interrupts to the driver.

`EXTRA` After *count* number of interrupts, start delivering *operand* number of extra interrupts for the next *failcount* number of real interrupts.

The default value for *operand* and *operator* is to corrupt the data access by flipping each bit (XOR with -1).

`-f` *acc_chk*

If the *acc_chk* parameter is set to 1 or `pio`, then the driver's calls to `ddi_check_acc_handle(9F)` return `DDI_FAILURE` when the access count goes to 1. If the *acc_chk* parameter is set to 2 or `dma`, then the driver's calls to `ddi_check_dma_handle(9F)` return `DDI_FAILURE` when the access count goes to 1.

-w *max_wait_period* [*report_interval*]

Constrain the period for which an error definition will remain active. The option applies only to non-logging errdefs. If an error definition remains active for *max_wait_period* seconds, the test will be aborted. If *report_interval* is set to a nonzero value, the current status of the error definition is reported to standard output every *report_interval* seconds. The default value is zero. The status of the errdef is reported in parsable format (eight fields, each separated by a colon (:) character, the last of which is a string enclosed by double quotes and the remaining seven fields are integers):

ft:*mt*:*ac*:*fc*:*chk*:*ec*:*s*:*"message"* which are defined as follows:

ft	The UTC time when the fault was injected.
mt	The UTC time when the driver reported the fault.
ac	The number of remaining non-faulting accesses.
fc	The number of remaining faulting accesses.
chk	The value of the *acc_chk* field of the errdef.
ec	The number of fault reports issued by the driver against this errdef (*mt* holds the time of the initial report).
s	The severity level reported by the driver.
"message"	Textual reason why the driver has reported a fault.

-h

Display the command usage string.

-s *collect_time*

If *acc_types* is given with the -a option and includes log, the errdef will log accesses for *collect_time* seconds (the default is to log until the log becomes full). Note that, if the errdef

specification matches multiple driver handles, multiple logging errdefs are registered with the `bofi` driver and logging terminates when all logs become full or when *collect_time* expires or when the associated errdefs are cleared. The current state of the log can be checked with the `th_manage(1M)` command, using the `broadcast` parameter. A log can be terminated by running `th_manage(1M)` with the `clear_errdefs` option or by sending a `SIGALRM` signal to the `th_define` process. See `alarm(2)` for the semantics of `SIGALRM`.

`-p` *policy*

Applicable when the *acc_types* option includes `log`. The parameter modifies the policy used for converting from logged accesses to errdefs. All policies are inclusive:

- Use `rare` to bias error definitions toward rare accesses (default).
- Use `operator` to produce a separate error definition for each operator type (default).
- Use `common` to bias error definitions toward common accesses.
- Use `median` to bias error definitions toward median accesses.
- Use `maximal` to produce multiple error definitions for duplicate accesses.
- Use `unbiased` to create unbiased error definitions.
- Use `onebyte`, `twobyte`, `fourbyte`, or `eightbyte` to select errdefs corresponding to 1, 2, 4 or 8 byte accesses (if chosen, the `-xr` option is enforced in order to ensure that `ddi_rep_*()` calls are decomposed into *multiple single accesses*).
- Use `multibyte` to create error definitions for multibyte accesses performed using `ddi_rep_get*()` and `ddi_rep_put*()`.

Policies can be combined by adding together these options. See the NOTES section for further information.

-x *flags*

Applicable when the *acc_types* option includes `log`. The *flags* parameter modifies the way in which the `bofi` driver logs accesses. It is specified as a string containing any combination of the following letters:

`w` Continuous logging (that is, the log will wrap when full).

`t` Timestamp each log entry (access times are in seconds).

`r` Log repeated I/O as individual accesses (for example, a `ddi_rep_get16(9F)` call which has a repcount of *N* is logged *N* times with each transaction logged as size 2 bytes. Without this option, the default logging behavior is to log this access once only, with a transaction size of twice the *repcount*).

-C *comment_string*

Applicable when the *acc_types* option includes `log`. It provides a comment string to be placed in any generated test scripts. The string must be enclosed in double quotes.

-e *fixup_script* [*args*]

Applicable when the *acc_types* option includes `log`. The output of a logging errdefs is to generate a test script for each driver access handle. Use this option to embed a command in the resulting script before the errors are injected. The generated test scripts will take an instance offline and bring it back online before injecting errors in order to bring the instance into a known fault–free state. The executable *fixup_script* will be called twice with the set of optional *args*— once just before the instance is taken offline and again after the instance has been brought online. The following variables are passed into the environment of the called executable:

`DRIVER_PATH` Identifies the device path of the instance.

`DRIVER_INSTANCE` Identifies the instance number of the device.

DRIVER_UNCONFIGURE	Has the value 1 when the instance is about to be taken offline.
DRIVER_CONFIGURE	Has the value 1 when the instance has just been brought online.

Typically, the executable ensures that the device under test is in a suitable state to be taken offline (unconfigured) or in a suitable state for error injection (for example configured, error free and servicing a workload). A minimal script for a network driver could be:

```
#!/bin/ksh

driver=xyznetdriver
ifnum=$driver$DRIVER_INSTANCE

if [[ $DRIVER_CONFIGURE = 1 ]]; then
    ifconfig $ifnum plumb
    ifconfig $ifnum ...
    ifworkload start $ifnum
elif [[ $DRIVER_UNCONFIGURE = 1 ]]; then
    ifworkload stop $ifnum
    ifconfig $ifnum down
    ifconfig $ifnum unplumb
fi
exit $?
```

The `-e` option must be the last option on the command line.

If the `-a log` option is selected but the `-e` option is not given, a default script is used. This script repeatedly attempts to detach and then re-attach the device instance under test.

使用例

Examples of Error Definitions

```
th_define -n foo -i 1 -a log
```

Logs all accesses to all handles used by instance 1 of the `foo` driver while running the default workload (attaching and detaching the instance). Then generates a set of test scripts to inject appropriate errdefs while running that default workload.

```
th_define -n foo -i 1 -a log pio
```

Logs PIO accesses to each PIO handle used by instance 1 of the `foo` driver while running the default workload (attaching and detaching the instance). Then generates a set of test scripts to inject appropriate errdefs while running that default workload.

```
th_define -n foo -i 1 -p onebyte median -e fixup arg -now
```

Logs all accesses to all handles used by instance 1 of the `foo` driver while running the workload defined in the fixup script `fixup` with arguments `arg` and `-now`. Then generates a set of test scripts to inject appropriate errdefs while running that workload. The resulting error definitions are requested to focus upon single byte accesses to locations that are accessed a *median* number of times with respect to frequency of access to I/O addresses.

```
th_define -n se -l 0x20 1 -a pio_r -o OR 0x4 -c 10 1000
```

Simulates a stuck serial chip command by forcing 1000 consecutive read accesses made by any instance of the `se` driver to its command status register, thereby returning status busy.

```
th_define -n foo -i 3 -r 1 -a pio_r -c 0 1 -f 1 -o OR 0x100
```

Causes 0x100 to be ORed into the next physical I/O read access from any register in register set 1 of instance 3 of the `foo` driver. Subsequent calls in the driver to `ddi_check_acc_handle()` return `DDI_FAILURE`.

```
th_define -n foo -i 3 -r 1 -a pio_r -c 0 1 -o OR 0x0
```

Causes 0x0 to be ORed into the next physical I/O read access from any register in register set 1 of instance 3 of the `foo` driver. This is of course a no-op.

```
th_define -n foo -i 3 -r 1 -l 0x8100 1 -a pio_r -c 0 10 -o EQ 0x70003
```

Causes the next ten next physical I/O reads from the register at offset 0x8100 in register set 1 of instance 3 of the `foo` driver to return 0x70003.

```
th_define -n foo -i 3 -r 1 -l 0x8100 1 -a pio_w -c 100 3 -o AND
0xffffffffffffefff
```

The next 100 physical I/O writes to the register at offset 0x8100 in register set 1 of instance 3 of the `foo` driver take place as normal. However, on each of the three subsequent accesses, the 0x1000 bit will be cleared.

```
th_define -n foo -i 3 -r 1 -l 0x8100 0x10 -a pio_r -c 0 1 -f 1 -o XOR 7
```

Causes the bottom three bits to have their values toggled for the next physical I/O read access to registers with offsets in the range 0x8100 to 0x8110 in register set 1 of instance 3 of the `foo` driver. Subsequent calls in the driver to `ddi_check_acc_handle()` return `DDI_FAILURE`.

```
th_define -n foo -i 3 -a pio_w -c 0 1 -o NO 0
```

Prevents the next physical I/O write access to any register in any register set of instance 3 of the `foo` driver from going out on the bus.

```
th_define -n foo -i 3 -l 0 8192 -a dma_r -c 0 1 -o OR 7
```

Causes 0x7 to be ORed into each `long long` in the first 8192 bytes of the next DMA read, using any DMA handle for instance 3 of the `foo` driver.

```
th_define -n foo -i 3 -r 2 -l 0 8 -a dma_r -c 0 1 -o OR 0x7070707070707070
```

Causes 0x70 to be ORed into each byte of the first `long long` of the next DMA read, using the DMA handle with sequential allocation number 2 for instance 3 of the `foo` driver.

```
th_define -n foo -i 3 -l 256 256 -a dma_w -c 0 1 -f 2 -o OR 7
```

Causes 0x7 to be ORed into each `long long` in the range from offset 256 to offset 512 of the next DMA write, using any DMA handle for instance 3 of the `foo` driver. Subsequent calls in the driver to `ddi_check_dma_handle()` return `DDI_FAILURE`.

```
th_define -n foo -i 3 -r 0 -l 0 8 -a dma_w -c 100 3 -o AND 0xffffffffffffefff
```

The next 100 DMA writes using the DMA handle with sequential allocation number 0 for instance 3 of the `foo` driver take place as normal. However, on each of the three subsequent accesses, the 0x1000 bit will be cleared in the first `long long` of the transfer.

```
th_define -n foo -i 3 -a intr -c 0 6 -o LOSE 0
```

Causes the next six interrupts for instance 3 of the `foo` driver to be lost.

```
th_define -n foo -i 3 -a intr -c 30 1 -o EXTRA 10
```

When the thirty-first subsequent interrupt for instance 3 of the `foo` driver occurs, a further ten interrupts are also generated.

```
th_define -n foo -i 3 -a intr -c 0 1 -o DELAY 1024
```

Causes the next interrupt for instance 3 of the `foo` driver to be delayed by 1024 microseconds.

注意事項

The policy option in the `th_define -p` syntax determines how a set of logged accesses will be converted into the set of error definitions. Each logged access will be matched against the chosen policies to determine whether an error definition should be created based on the access.

Any number of policy options can be combined to modify the generated error definitions.

Bytewise Policies

These select particular I/O transfer sizes. Specifing a byte policy will exclude other byte policies that have not been chosen. If none of the byte type policies is selected, all transfer sizes are treated equally. Otherwise, only those specified transfer sizes will be selected.

`onebyte`	Create errdefs for one byte accesses (`ddi_get8()`)
`twobyte`	Create errdefs for two byte accesses (`ddi_get16()`)
`fourbyte`	Create errdefs for four byte accesses (`ddi_get32()`)
`eightbyte`	Create errdefs for eight byte accesses (`ddi_get64()`)
`multibyte`	Create errdefs for repeated byte accesses (`ddi_rep_get*()`)

Frequency of Access Policies

The frequency of access to a location is determined according to the access type, location and transfer size (for example, a two-byte read access to address A is considered distinct from a four-byte read access to address A). The algorithm is to count the number of accesses (of a given type and size) to a given location, and find the locations that were most and least accessed (let *maxa* and *mina* be the number of times these locations were accessed, and *mean* the total number of accesses divided by total number of locations that were accessed). Then a rare access is a location that was accessed less than

(mean - mina) / 3 + mina

times. Similarly for the definition of common accesses:

maxa - (maxa - mean) / 3

A location whose access patterns lies within these cutoffs is regarded as a location that is accessed with median frequency.

`rare`	Create errdefs for locations that are rarely accessed.
`common`	Create errdefs for locations that are commonly accessed.
`median`	Create errdefs for locations that are accessed a median frequency.

Policies for Minimizing errdefs

If a transaction is duplicated, either a single or multiple errdefs will be written to the test scripts, depending upon the following two policies:

`maximal`	Create multiple errdefs for locations that are repeatedly accessed.
`unbiased`	Create a single errdef for locations that are repeatedly accessed.
`operators`	For each location, a default operator and operand is typically applied. For maximal test coverage, this default may be modified using the `operators` policy so that a separate errdef is created for each of the possible corruption operators.

関連項目

`kill(1)`, `th_manage(1M)`, `alarm(2)`, `ddi_check_acc_handle(9F)`, `ddi_check_dma_handle(9F)`

名前

th_manage – manage the fault injection test harness

形式

`th_manage` *name instance command*

`th_manage` *path command*

機能説明

`th_manage` applies the action specified by *command* to the instance specified by *instance* of the driver specified by *name* (or the driver instance specified by *path*). The driver instance must be running fault injection specifications (errdefs) defined by `th_define(1M)`.

`th_manage` supports several commands that operate on the driver instance specified by *name* and *instance* (or *path*). The commands are:

`broadcast`	Awaken all `th_define` processes, causing them to display their current status and exit if the errdef is now defunct (that is, if *count*, *failcount* , and *acc_chk* are all zero).
`clear_acc_chk`	Awaken all `th_define` processes. If *count* and *failcount* are already zero, then set *acc_chk* to zero, so that `th_define` exits once it has displayed its status.
`clear_errdefs`	Awaken all `th_define processes`. *count*, *failcount* and *acc_chk* are all set to zero so that all `th_define` commands exit once they have displayed their status.
`clear_errors`	Awaken all `th_define` processes. If *count* is already zero, set *failcount* and *acc_chk* to zero, so that `th_define` exits once it has displayed its status.
`get_handles`	List all the access handles.
`start`	Begin or resume execution of all errdefs.
`stop`	Suspend all errdefs for this *name* and *instance* (or *path).*

使用例

例 1　Useful Commands

To begin the tests, enter:

```
# th_manage foo 0 start
```

To check the status of the errdefs, enter:

```
# th_manage foo 0 broadcast
```

This causes each `th_define` process to print out its current status.

If the driver has reported a fatal error, you can take the driver offline using `libdevice`, clear the error condition by entering:

```
# th_manage foo 0 clear_acc_chk
```

例 1 Useful Commands (続き)

or

```
# th_manage foo 0 clear_errors
```

and bring the driver online again using `libdevice`.

To terminate testing, enter:

```
# th_manage foo 0 clear_errdefs
```

関連項目

th_define(1M)

名前

tic – terminfo compiler

形式

`tic [-v [n]] [-c] file`

機能説明

The command `tic` translates a `terminfo` file from the source format into the compiled format. The results are placed in the directory `/usr/share/lib/terminfo`. The compiled format is necessary for use with the library routines in `curses(3CURSES)`.

If the environment variable `TERMINFO` is set, the compiled results are placed there instead of `/usr/share/lib/terminfo`.

Total compiled entries cannot exceed 4096 bytes. The name field cannot exceed 128 bytes. Terminal names exceeding 14 characters will be truncated to 14 characters and a warning message will be printed.

オプション

The following options are supported:

`-c` Specifies to check only `file` for errors. Errors in `use=` links are not detected.

`-v`[*n*] Specify that (verbose) output be written to standard error trace information showing `tic`'s progress. The optional integer *n* is a number from 1 to 10, indicating the desired level of detail of information. If *n* is omitted, the default level is 1. If *n* is specified and greater than 1, the level of detail is increased.

オペランド

`file` Contains one or more `terminfo` terminal descriptions in source format [see `terminfo(4)`]. Each description in the file describes the capabilities of a particular terminal. When a `use=`*entry-name* field is discovered in a terminal entry currently being compiled, `tic` reads in the binary from `/usr/share/lib/terminfo` to complete the entry. (Entries created from `file` will be used first. If the environment variable `TERMINFO` is set, that directory is searched instead of `/usr/share/lib/terminfo`.) `tic` duplicates the capabilities in *entry-name* for the current entry, with the exception of those capabilities that are explicitly defined in the current entry.

ファイル

`/usr/share/lib/terminfo/?/*` Compiled terminal description database

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`captoinfo(1M)`, `infocmp(1M)`, `curses(3CURSES)`, `terminfo(4)`, `attributes(5)`

注意事項

When an entry, for example, `entry_name_1`, contains a `use=`*entry_name_2* field, any canceled capabilities in *entry_name_2* must also appear in `entry_name_1` before `use=` for these capabilities to be canceled in `entry_name_1`.

名前

tncfg – configure trusted networking properties

形式

```
tncfg [-t template] [-e|-S [files|ldap]] [subcommand]
tncfg [-t template] [-e|-S [files|ldap]] -f command_file
tncfg -z zone [-e] [subcommand]
tncfg -z zone [-e] -f command_file
tncfg help
```

機能説明

The `tncfg` utility creates, modifies, and displays the configuration of various networking properties related to Trusted Extensions. The command requires that the SMF service, `svc:/system/labeld` is enabled. It can be executed only in the global zone.

Templates

A template is a collection of network security properties that define the rules for labeling packets received from remote hosts. Two host types are supported: `cipso` and `unlabeled`. Each template must specify one of these two host types. Hosts that are trusted to specify their own labels are assigned to `cipso` templates. Otherwise, hosts may be assigned to `unlabeled` templates, in which a single default label must be specified.

Hosts can be specified using hostnames, IP addresses, or masks. When masks are used, a prefix length that specifies how many bits are required for a match must be appended . Hosts cannot be assigned to more than one template. When masks are used, the entry with the longest matching prefix is used to associate a host with a template. Packets from hosts without a matching template are dropped.

Each template must include an upper and lower bound specifying the accreditation range of accepted labels. Additionally, up to four auxiliary labels can be specified to enumerate labels outside of this range. Services bound to multilevel ports can accept packets from hosts whose labels are within the accreditation range, or match one of the auxiliary labels.

Normally, the template settings and their corresponding hosts are persistently maintained in local files or by means of an LDAP directory, depending on the `-S` option. These settings are automatically loaded into kernel memory when the user commits the updates. If the `-e` (ephemeral) option is specified, only the current in-memory properties are displayed and updated. However, the list of hosts associated with an in-memory template is generally incomplete. To view the matching template for a specific host, use the `get` subcommand.

By default, an unlabeled template, `admin_low`, is installed with a default label of `ADMIN_LOW`, and two mask entries matching any IPv4 or IPv6 address, so that the global zone is initially able to contact any `unlabeled` hosts. It is recommended to remove these two mask entries once your network security policy is established. An additional template, `cipso`, is installed with no matching hosts. By default all local IP addresses are implicitly associated with this template, but it is recommended that they should be explicitly added to this or a new `cipso`-type template.

Searches for template and host entries are resolved in the order specified by means of the name service configuration file, `/etc/nsswitch.conf`. The keywords, `tnrhdb` and `tnrhtp`, are used to specify the search order for hosts and templates, respectively. Both the `files` and `ldap` repositories are supported, but it is recommended to specify `files` first.

Creating or modifying a template requires the authorization `solaris.label.network.manage`, which is included in the Object Label Management rights profile.

Zones

Zones are isolated execution environments described in `zones(5)`. Trusted Extensions requires a zone brand called `labeled` to which special properties apply. Each labeled zone must have a unique label property at which it executes processes. This is also the label at which it will accept packets from remote hosts for services bound to single-level ports. Explicit multilevel port can also be specified. Services with the privilege `net_bindmlp` can bind to these ports, and accept packets within the accreditation range or the auxiliary label set associated with the remote host.

Non-global zones must be configured using `zonecfg(1M)` prior to configuring these properties. In general, updates to each zone's properties, including the global zone, are applied when it is booted. However, the multilevel port properties of running zones are reloaded into kernel memory when updates are committed. If the `-e` (ephemeral) option is specified, the zone must be in the ready or running state, and only its multilevel port properties can be updated.

Creating or modifying a zone's trusted networking properties requires the authorization `solaris.label.zone.manage`, which is included in the Object Label Management rights profile.

Properties

The set of valid properties depends on whether the `-t` or `-z` option was used. The two sets are referred to as the template context and the zone context.

Only a single property value can be specified at a time. Values containing white space must be quoted. An equal sign is required between the property and its value.

The values that can be specified in the template context properties are described below.

`name=`*template_name*

The initial value for the name is specified using `-t` option using the command line. If the name is changed, the current template properties are applied to the newly named template. In this way an existing template can be cloned for subsequent editing. However, to avoid conflicts, the host entries from the initial template are not copied to the new template. The specified name must not match an existing template.

`host_type=cipso|unlabeled`

When the `unlabeled` host type is used, the value specified using the the `def_label` property is implicitly applied to the received packets. The `cipso` host type is used for hosts that are trusted to explicitly label their packets. The default is `unlabeled`.

`def_label=`*sensitivity_label*
: The default label assigned to IP packets that are not explicitly labeled by means of `cipso` or IPsec.

`doi=`*integer*
: A positive integer specifying the Domain of Interpretation for the binary representation of the labels. The default is `1`.

`min_label=`*sensitivity_label*
: The minimum label in the accreditation range for IP packets that are accepted by multilevel services.

`max_label=`*sensitivity_label*
: The maximum label in the accreditation range for IP packets that are accepted by multilevel services.

`aux_label=`*sensitivity_label*
: Additional labels, outside of the accreditation range, for IP packets that are accepted by multilevel services. Up to four labels may be specified, using the `add` subcommand repetitively.

`host=`*hostname*|*IP address*[/*prefix*]
: A hostname or an IP address to which the template properties apply. For IP addresses, both IPv4 and IPv6 formats can be used, followed by an optional slash and prefix length specifying the number of bits to match again IP addresses. The IPv4 address `0.0.0.0` has an implied prefix length of zero, and matches any IPv4 address. Multiple host values can be specified, using the `add` subcommand repetitively. There is no specific limit.

The values that may be specified in the zone context properties are described below.

`name=`*zone_name*
: The name of the zone, which must have previously been configured using `zonecfg(1M)`. The initial value for the name is specified using `-z` option on the command line. If the name is changed, the current zone properties are applied to the newly named zone. In this way an existing template can be cloned for subsequent editing. However, to avoid conflicts, the initial label value is not copied to the new zone configuration. The specified name must correspond to an existing zone without a trusted networking configuration.

`label=`*sensitivity_label*
: The sensitivity label of the zone. It must be unique for each zone. The global zone value must be `admin_low`.

`visible=yes|no`
: Specifies whether the zone responds to ping requests from hosts whose labels don't match the zone's label. The default is `no`.

`mlp_private=`*port*|[-*port2*]/`tcp|udp`
: A single port number, or a range of ports that privileged services can bind to and then accept requests from clients whose labels are with the accreditation range or the set of

auxiliary labels specified in their matching templates. The port specification must be followed by a a protocol, either `tcp` or `udp`. This value applies to all interfaces that are private to the zone. Multiple `mlp_private` values can be specified, using the `add` subcommand repetitively. This is only limited by the number of available ports.

`mlp_shared=`*port*|[-*port2*]/`tcp`|`udp`
A single port number, or a range of ports that privileged services can bind to and then accept requests from clients whose labels are with the accreditation range or the set of auxiliary labels specified in their matching templates. The port specification must be followed by a protocol, either `tcp` or `udp`. This value applies to any `all-zones` interfaces, and must not overlap with the `mlp_shared` ports specifications for other zones. Multiple `mlp_shared` values can be specified, using the `add` subcommand repetitively. This is only limited by the number of available ports.

オプション

The following options are supported:

`-e`
Specifies that the data is ephemeral, affecting only what is currently loaded into kernel memory.

`-f` *command_file*
Specifies the name of `tncfg` command file. *command_file* is a text file of `tncfg` subcommands, one per line.

`-t` *template*
Specifies the template name. If the named template does not exist, a new template is created. If neither `-t` nor `-z` is specified, the template context is assumed using `cipso` as the default template name.

`-S` *repository*
The valid repositories are `files` and `ldap`. The repository specifies which name service will be updated. The default repository is files.

`-z` *zone*
Specifies the zone name. The zone must have previously been configured by means of `zonecfg(1M)`.

サブコマンド

Subcommands can be provided on the command line or interactively. Multiple subcommands, separated by semicolons, can be specified on the command line by enclosing the entire set in quotation marks. The lack of subcommands implies an interactive session, during which auto-completion of subcommands can be invoked using the tab key.

The add, clear, and remove subcommands are used for properties that can accept multiple values. However, only one value can be specified at a time.

Subcommands which can result in destructive actions or loss of work have an `-F` option to force the action. If input is from a terminal device, the user is prompted when appropriate if such a subcommand is given without the `-F` option. Otherwise, the action is disallowed, with a diagnostic message written to standard error.

The following subcommands are supported:

`add` *property-name=property-value*
: Adds the specified value to the current property values. This subcommand can only be applied to properties that accept multiple values. Use the `set` subcommand for single-value properties.

`clear` *property-name*
: Clears all of the values for the property. Only those properties that accept multiple assignments, using the `add` subcommand, can be cleared.

`commit`
: Commits the current configuration from memory to stable storage and into the kernel. The configuration must be committed for the changes to take effect. Until the in-memory configuration is committed, you can remove changes with the `revert` subcommand. The commit operation is attempted automatically upon completion of a `tncfg` session. Since a configuration must be correct to be committed, this operation automatically does a `verify`.

`delete` [`-F`]
: Deletes the specified template or zone configuration from the current name service.

  Specify the `-F` option to force the action. If the deletion is allowed, its action is instantaneous and the session is terminated.

`export` [`-f` *output-file*]
: Displays configuration to standard output. Use the `-f` option to display the configuration to *output-file*. This option produces output in a form suitable for use in a command file.

`get` host=*hostname* | *IP address*[/*prefix*]
: Displays the template name corresponding to the specified host using the kernel's in-memory mapping.

`help` [`usage`] [`subcommands`] [`properties`] [*subcommand*] [*property*]
: Displays general help or help about a given topic.

`info` *property-name*
: Displays information about the current template or zone, or the specified property in a parseable format.

`list`
: Lists the names of the templates or zones that have been configured.

`remove` *property-name=property-value*
: Removes the specified value from the property. Only those properties that accept multiple assignments, using the `add` subcommand, can be removed.

`set` *property-name=property-value*
: Sets a given property name to the given value. Properties that can take multiple values are assigned using the `add` subcommand, instead of `set`.

`verify`
Verifies the current configuration for correctness:

- The required properties are specified;
- the values are valid for each key word;
- the user is authorized to specify the values.

`revert` [`-F`]
Causes the configuration to revert to the last committed state. The `-F` option can be used to force the action.

`exit` [`-F`]
Exits the `tncfg` session. A `commit` is automatically attempted if needed. You can also use an EOF character to exit `tncfg`. The `-F` option can be used to force the action.

使用例

例 1 Using the info Subcommand

The command below displays the properties of a `cipso` template are displayed. The subcommand is specified on the command line.

```
example% tncfg -t cipso info
             name=cipso
             host_type=cipso
             doi=1
             min_label=ADMIN_LOW
             max_label=ADMIN_LOW
             host=10.5.233.74
```

例 2 Using the export Subcommand

The following example shows an interactive session that exports the configuration of a zone in a format the could be imported to another machine with an equivalent zone.

```
example% tncfg -t public
tncfg:public> export
set name=public
set host_type=unlabeled
set doi=1
set def_label="PUBLIC"
set min_label="PUBLIC"
set max_label="CONFIDENTIAL : NEED TO KNOW"
add aux_label="SANDBOX PLAYGROUND"
add host=myserver.oracle.com
add host=10.5.0.0/16
tncfg:public> exit
```

例 3 Assigning Properties to a Zone

In the following example, the public zone is configured to be a multi-level NFS server.

例 3 Assigning Properties to a Zone (続き)

```
example% tncfg -z public
tncfg:public> info
     name=public
     label=PUBLIC
     visible=no
tncfg:public> add mlp_private=111/tcp
tncfg:public> add mlp_private=111/ucp
tncfg:public> add mlp_private=2049/tcp
tncfg:public> commit
tncfg:public> exit
```

終了ステータス

0
: Successful completion.

1
: An error occurred.

ファイル

- /etc/security/tsol/tnrhtp
- /etc/security/tsol/tnrhdb
- /etc/security/tsol/tnzonecfg

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/trusted
Interface Stability	See below.

The invocation and subcommands are committed. Output, except for the export and info subcommands, is Not-an-Interface.

関連項目

tnctl(1M), tnd(1M), tninfo(1M), txzonemgr(1M), zonecfg(1M), nsswitch.conf(4), attributes(5), labels(5), zones(5)

注意事項

The Labeled Zone Manager, txzonemgr(1M), is an alternative application for configuring Trusted Extensions. It invokes the tncfg command internally, and provides an interactive GUI-based user interface.

名前

tnchkdb – トラステッドネットワークデータベースのファイル構文の検査

形式

```
/usr/sbin/tnchkdb  [-h path] [-t path] [-z path]
```

機能説明

tnchkdb は、tnrhtp、tnrhdb、および tnzonecfg データベースの構文を検査します。デフォルトでは、各ファイルのパスは次のとおりです。

- /etc/security/tsol/tnrhtp
- /etc/security/tsol/tnrhdb
- /etc/security/tsol/tnzonecfg

コマンド行で -h (tnrhdb)、-t (tnrhtp)、および -z (tnzonecfg) オプションとともにパスを指定することで、一部またはすべてのファイルの代替パスを指定できます。このオプションは、変更したファイルセットを新規システムデータベースとしてインストールする前にテストする場合に便利です。

3 つのデータベースファイルすべての整合性が検査されます。すべてのファイルが構文的に正しく、かつ可能な範囲内で意味的に正しい場合、tnchkdb は終了ステータス 0 を返します。1 つ以上のファイルにエラーが存在する場合、終了ステータス 1 が返されます。ファイルを読み込むことができないなど、コマンド行の指定に問題がある場合は、終了ステータス 2 が返されます。エラーは、標準エラーに書き込まれます。

tnrhtp 内にエラーが存在する場合、エラーが階層的に発生することを防ぐために、tnrhdb 内のテンプレート名は検査されません。

tnchkdb は任意のラベルで実行できますが、標準の /etc/security/tsol ファイルを表示できるのは大域ゾーンだけです。

オプション

-h [*path*]　*path* を検査して、tnrhdb の構文が適切かどうかを調べます。*path* が指定されない場合は、/etc/security/tsol/tnrhdb を検査します。

-t [*path*]　*path* を検査して、tnrhtp の構文が適切かどうかを調べます。*path* が指定されない場合は、/etc/security/tsol/tnrhtp を検査します。

-z [*path*]　*path* を検査して、tnzonecfg の構文が適切かどうかを調べます。*path* が指定されない場合は、/etc/security/tsol/tnzonecfg を検査します。

使用例

例 1　エラーメッセージの例

tnchkdb コマンドは、CIPSO エラーを検査します。この例では、admin_low テンプレートのデフォルトラベルが、不正な値 ADMIN_HIGH です。

```
# tnchkdb
checking /etc/security/tsol/tnrhtp ...
tnchkdb: def_label classification 7fff is invalid for cipso labels:
line 14 entry admin_low
tnchkdb: def_label compartments 241-256 must be zero for cipso labels:
line 14 entry admin_low
```

例 1 エラーメッセージの例 (続き)

```
checking /etc/security/tsol/tnrhdb ...
checking /etc/security/tsol/tnzonecfg ...
```

ファイル

/etc/security/tsol/tnrhdb	トラステッドネットワーク遠隔ホストデータベース
/etc/security/tsol/tnrhtp	トラステッドネットワーク遠隔ホストテンプレート
/etc/security/tsol/tnzonecfg	トラステッドゾーン構成データベース

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

このコマンド行は「確実」です。出力は「不確実」です。

関連項目

tnd(1M), tnctl(1M), attributes(5)

『Trusted Extensions 構成と管理』

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

LDAP を使って欠落したテンプレートを提供する場合、tnrhtp および tnrhdb の構成が一貫してはいなくても有効であることがあります。

名前

tnctl – Trusted Extensions ネットワークパラメータの構成

形式

```
/usr/sbin/tnctl [-dfv] [-h host [/prefix] [:template]]
      [-m zone:mlp:shared-mlp][-t template [:key=val [;key=val]]]
      [-HTz] file]
```

機能説明

tnctl は、Solaris カーネル内のトラステッドネットワークパラメータを操作するためのインタフェースを提供します。

システムの起動時に、Solaris Trusted Extensions の初期化処理の一部として、smf(5) スクリプトにより tnctl が大域ゾーン内で実行されます。tnctl は、通常のシステム管理での使用を意図したコマンドではありません。代わりに、ローカルのトラステッドネットワークデータベースファイルが変更された場合、管理者は最初に tnchkdb(1M) を発行して構文を検査し、次にこのコマンドを使用してカーネルのコピーを更新します。

```
# svcadm restart svc:/network/tnctl
```

実行中のシステム上の遠隔ホストおよびテンプレート情報を変更することの危険については、「警告」を参照してください。

オプション

-d
一致するエントリをカーネルから削除します。デフォルトでは、新規エントリを追加します。

MLP を削除する場合は、MLP の範囲が正確に一致している必要があります。MLP の指定は、次の形式で行います。

port[-*port*]/*protocol*

ここで、*port* には 1 から 65535 の範囲の数値を指定できます。または、任意の既知のサービスを指定できます (services(4) を参照)。protocol には 1 から 255 の範囲の数値、または任意の既知のプロトコル (protocols(4) を参照) を指定できます。

-f
コマンド行で指定されたエントリをロードする前に、すべてのカーネルエントリをフラッシュします。1 つ以上のエントリの構文解析が成功しない場合、フラッシュは実行されません。

-v
冗長モードを有効にします。

-h *host*[/ *prefix*][:*template*]
指定された *host* のローカルホスト上のカーネル遠隔ホストキャッシュを更新します。テンプレート名が指定されている場合は、指定された *template* を使ってカーネルのキャッシュを変更します。*prefix* が指定されていない場合、tnrhdb の解釈に使用した規則に従って、示された接頭辞の長さが決定されます。-d を指定する場合は、テンプレート名は指定できません。

-m *zone*: *mlp*:*shared-mlp*
指定された *zone* に対するカーネルのマルチレベルポート (MLP) 構成キャッシュを変更します。*zone* には、更新するゾーンを指定します。*mlp* および *shared-mlp* は、ゾーン固有および共有の IP アドレス用の MLP を指定します。*shared-mlp* フィールドは、大域ゾーンのみで有効です。

-t *template*[*key*=*val*[;*key*=*val*]]
template のカーネルテンプレートキャッシュを更新します。また、*key*=*val* ペアのリストが指定された場合は、カーネルのキャッシュを変更して指定されたエントリを使用します。-d が指定された場合は、*key*=*val* ペアは指定できません。

-T *file*
file 内のテンプレートエントリすべてをカーネルキャッシュに読み込みます。

-H *file*
file 内の遠隔ホストエントリすべてをカーネルキャッシュに読み込みます。

-z *file*
大域ゾーンの MLP だけを *file* からカーネルキャッシュに読み込みます。非大域ゾーンの MLP の再読み込みを行うには、ゾーンをリブートします。

```
# zoneadm -z non-global zone reboot
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	不確実

ファイル

/etc/security/tsol/tnrhdb	トラステッドネットワーク遠隔ホストデータベース
/etc/security/tsol/tnrhtp	トラステッドネットワーク遠隔ホストテンプレート
/etc/security/tsol/tnzonecfg	トラステッドゾーン構成データベース
/etc/nsswitch.conf	ネームサービススイッチの構成ファイル

関連項目

svcs(1), svcadm(1M), tninfo(1M), tnd(1M), tnchkdb(1M), zoneadm(1M), nsswitch.conf(4), protocols(4), services(4), attributes(5), smf(5)

『Trusted Extensions 構成と管理』

警告

ネットワークの稼働中にテンプレートを変更すると、不特定数のホストのセキュリティー表示が変更される可能性があります。

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

tnctl サービスは、サービス管理機能 smf(5) により、サービス識別子の下で管理されます。

```
svc:/network/tnctl
```

サービスのステータスは、svcs(1) を使って問い合わせることができます。このサービスに対する管理操作 (カーネルキャッシュの更新など) は、svcadm(1M) を使って実行できます。次に例を示します。

```
svcadm restart svc:/network/tnctl
```

名前

tnd – トラステッドネットワークデーモン

形式

`/usr/sbin/tnd [-p poll-interval]`

機能説明

tnd (トラステッドネットワークデーモン) は、トラステッドネットワークデータベースを使ってカーネルを初期化し、また要求に従って LDAP サーバーおよびローカルファイルからデータベースの再読み込みを行います。tnd は、構成データベースを読み込む際、nsswitch.conf(4) で指定された順序に従います。

tnd が大域ゾーンのブート処理中に svc:/network/tnd smf(5) サービスから起動されるように意図されているのは、システムが LDAP クライアントとして構成されている場合だけです。トラステッドネットワークデータベースとしてローカルファイルのみを使用するシステムは、tnd の代わりに tnctl(1M) を使用します。

tnd は次のデータベースをカーネルに読み込みます。遠隔ホストデータベース tnrhdb と遠隔ホストテンプレートデータベース tnrhtp です。tnd はさらに、関連する LDAP データベースやローカルデータベースに変更がないか定期的にスキャンし、その結果に応じてカーネルキャッシュを更新します。

ローカルのトラステッドネットワークデータベースファイルが変更された場合、管理者は tnchkdb(1M) を実行して構文を検査するとともに、svcadm refresh svc:/network/tnd を実行して tnd によるデータベーススキャンをすぐに開始するようにしてください。

tnd は smf(5) スクリプトから起動され、大域ゾーンで実行されることを前提としています。次の svcadm コマンドは、特定のアクションを実行するように tnd に指示します。

svcadm refresh svc:/network/tnd	ローカルおよび LDAP tnrhdb と tnrhtp データベースの再スキャンが開始されます。tnd は、検出された変更点を使ってカーネルデータベースを更新します。
svcadm disable svc:/network/tnd	tnd デーモンを停止します。カーネルデータベースは変更されません。

tnd がデバッグモードで実行されるかどうかは、次のサービス管理機能 (SMF) プロパティーの値で決まります。

```
tnd/debug_level = 0
```

前述のように値が 0 の場合、デバッグ情報は収集されません。1 の場合はデバッグが有効になります。デフォルト値は 0 です。デバッグ出力は /var/tsol/tndlog ログファイルに送信されます。

オプション

-p *poll-interval*	ポーリング間隔を *poll-interval* 秒に設定します。デフォルトの *poll-interval* は 1800 秒 (30 分) です。

使用例

例1 ポーリング間隔の変更

次のコマンドは、ポーリング間隔を 1 時間に変更して、この間隔を SMF リポジトリに格納します。次回のブート時に、tnd のポーリング間隔は 1 時間になります。

```
# svccfg -s network/tnd setprop tnd/poll_interval=3600
```

次のコマンドは、ポーリング間隔を変更しますが、リポジトリの更新は行いません。次回のブート時、tnd のポーリング間隔はデフォルトの 30 分のままです。

```
# tnd -p 3600
```

ファイル

/etc/security/tsol/tnrhdb	トラステッドネットワーク遠隔ホストデータベース
/etc/security/tsol/tnrhtp	トラステッドネットワーク遠隔ホストテンプレート
/etc/security/tsol/tnzonecfg	トラステッドゾーン構成データベース
/etc/nsswitch.conf	ネームサービススイッチの構成ファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

コマンドの呼び出しは「確実」です。サービスは「非公開」です。

関連項目

svcs(1), svcadm(1M), tninfo(1M), tnctl(1M), tnchkdb(1M), nsswitch.conf(4), attributes(5), smf(5)

『Trusted Extensions 構成と管理』

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

tnd サービスは、サービス管理機能 smf(5) により、サービス識別子の下で管理されます。

```
svc:/network/tnd
```

サービスのステータスは、svcs(1) を使って問い合わせることができます。このサービスに対する管理操作 (デーモンの再起動要求など) は、svcadm(1M) を使って実行できます。次に例を示します。

```
svcadm restart svc:/network/tnd
```

名前

tninfo – カーネルレベルのネットワークに関する情報と統計の出力

形式

`/usr/sbin/tninfo [-h hostname] [-m zone-name] [-t template]`

機能説明

`tninfo` は、カーネルレベルのネットワーク情報と統計を取得して表示するためのインタフェースを提供します。

オプション

`-h` *hostname*	遠隔ホストキャッシュ内にある、指定されたホストのセキュリティー構造を表示します。出力には、`tnrhdb` データベース内で指定された内容が反映されます。
`-m` *zone-name*	指定したゾーンに関連付けられた MLP 構成を表示します。出力には、`tnzonecfg` データベース内で指定された内容が反映されます。
`-t` *template*	指定した *template* に関連付けられている構造を表示します。出力には、`tnrhtp` データベース内で指定された内容が反映されます。

使用例

例 1 カーネル内にキャッシュされた遠隔ホスト構造を表示する

この例では、カーネル内にキャッシュされた遠隔ホスト構造を表示します。出力には、`tnrhdb` データベース内の定義が反映されます。

```
# tninfo -h machine1
   IP address= 192.168.8.61
   Template = cipso
```

例 2 大域ゾーンのマルチレベルポートを表示する

この例では、カーネルでキャッシュされた大域ゾーンの MLP を表示します。出力には、`tnzonecfg` データベース内の定義、および動的に割り当てられたすべての MLP が反映されます。`private` は、ゾーン固有の MLP を示します。

```
# tninfo -m global
private:23/tcp;111/tcp;111/udp;515/tcp;2049/tcp;6000-6003/tcp;
        32812/tcp;36698/ip;38634/tcp;64365/ip
shared: 6000-6003/tcp
```

例 3 cipso テンプレート定義を表示する

この例では、カーネルでキャッシュされた `cipso` テンプレート定義を表示します。出力には、`tnrhtp` データベース内の定義が反映されます。

```
# tninfo -t cipso
=====================================
   Remote Host Template Table Entries:

   ___________________________
   template: cipso
   host_type: CIPSO
   doi: 1
   min_sl: ADMIN_LOW
   hex: ADMIN_LOW
```

例 3　cipso テンプレート定義を表示する　　(続き)

```
max_sl: ADMIN_HIGH
hex: ADMIN_HIGH
```

ファイル

/etc/security/tsol/tnrhdb	トラステッドネットワーク遠隔ホストデータベース
/etc/security/tsol/tnrhtp	トラステッドネットワーク遠隔ホストテンプレート
/etc/security/tsol/tnzonecfg	トラステッドゾーン構成データベース

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/trusted
インタフェースの安定性	下記を参照。

このコマンド行は「確実」です。出力は「不確実」です。

関連項目

tnd(1M), tnctl(1M), attributes(5)

『Trusted Extensions 構成と管理』

注意事項

このマニュアルページに記載されている機能を利用できるのは、システムが Trusted Extensions を使用して構成されている場合だけです。

名前

tpmadm – トラステッドプラットフォームモジュールの管理

形式

```
tpmadm status

tpmadm init

tpmadm clear [owner | lock]

tpmadm auth

tpmadm keyinfo [uuid]

tpmadm deletekey uuid

tpmadm pcrextend pcr [filename]

tpmadm pcrreset pcr
```

機能説明

トラステッドプラットフォームモジュール (TPM) は、鍵の保護と格納、およびオペレーティングシステムの起動に使用されるソフトウェアの信頼性の測定に対応したハードウェアコンポーネントです。`tpmadm` ユーティリティーは、オペレーティングシステムやその他のプログラムから TPM を使用できるように、TPM の初期化や管理を行うために使用します。

TPM サブシステムは、オペレーティングシステムやユーザーによって使用される鍵を、任意の数だけ格納して管理できます。各鍵は汎用一意識別子 (UUID) によって識別されます。

TPM は一度に限られた数の鍵しか保持できませんが、サポートソフトウェアによって、必要に応じて鍵の読み込みや読み込み解除が自動的に行われます。TPM の外部に格納される鍵は必ずその親鍵によって暗号化つまり「ラップ」されるため、その鍵が TPM の外部で読み取り可能な形式で公開されることは決してありません。

TPM を使用するには、プラットフォームの所有者が TPM を事前に初期化しておく必要があります。この処理では、特権操作の承認時に使用される、所有者のパスワードを設定します。

TPM 所有者は従来のスーパーユーザーに似ていますが、両者の間には 2 つの重要な違いがあります。まず、プロセス特権は、TPM 機能にアクセスする際は重要ではありません。呼び出し元プロセスの特権レベルにかかわらず、すべての特権操作で所有者のパスワードを把握しておくことが必要になります。第 2 に、TPM 所有者は、TPM 鍵で保護されたデータのアクセス制御を無効にすることはできません。所有者は、TPM を初期化し直すことで結果的にデータを破棄することはできますが、ほかのユーザーが所有する TPM 鍵で暗号化されたデータにアクセスすることはできません。

サブコマンド

後述の各サブコマンドは、次の形式で使用されます。

```
# tpamadm <subcommand> [operand]
```

`status`
TPM に関するステータス情報を報告します。出力には、TPM の所有権が確立されたかどうか、PCR の現在の内容、および TPM リソース (通信セッションや読み込まれた鍵など) の使用状況に関する基本情報が含まれます。

`init`
TPM を初期化して使用可能な状態にします。その際、所有者の承認パスワードを設定することで TPM の所有権が取得されます。TPM の所有権を取得すると、この TPM によって作成されるすべての鍵の祖先となる新しいストレージルート鍵が作成されます。このコマンドの発行後に再度初期化を行うには、まず BIOS 操作を使用して TPM をリセットする必要があります。

`auth`
TPM の所有者の承認パスワードを変更します。

`clear lock`
失敗した認証試行の回数をクリアします。失敗した認証試行の回数が多くなると、その後の試行に対する TPM の応答速度が遅くなりますが、これは、総当たり検索で所有者のパスワードを発見しようとする行為の邪魔をするためです。このコマンドは、所有者の正しいパスワードを要求してから、失敗試行の回数をリセットします。

`clear owner`
TPM を無効にし、TPM を所有されていない状態に戻します。この処理は、現在の TPM 所有者の正しいパスワードを要求してから、TPM に関連付けられたすべての鍵とデータを無効にします。TPM を再度使用するには、システムを再起動し、BIOS または ILOM ブート前環境から TPM を再度有効化し、`tpmadm init` コマンドを使用して TPM を再度初期化する必要があります。

`keyinfo` [*uuid*]
TPM サブシステムに格納された鍵に関する情報を報告します。追加の引数が指定されなかった場合、このサブコマンドはすべての鍵の簡易一覧を生成します。特定の鍵の UUID が指定された場合は、その鍵の詳細情報が表示されます。

`deletekey` *uuid*
指定された UUID の鍵を、TPM サブシステムの永続ストレージから削除します。

`pcrextend` *pcr* [*filename*]
filename の内容の SHA-1 ハッシュを作成し、そのハッシュ値を拡張対象データとして使用して、指定された PCR で PCR 拡張操作を実行します。ファイル名が指定されていない場合、データは標準入力から読み込まれます。

`pcrreset` *pcr*
指定された PCR を初期状態 (すべてゼロ) にリセットします。

終了ステータス

要求された処理の完了後に、tpmadm は次のいずれかのステータス値を返して終了します。

0
正常終了。

1
失敗。要求された操作を完了できませんでした。

2
使用方法のエラー。tpmadm コマンドの呼び出し時に無効な引数が指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

prof_attr(4), attributes(5)

SUNWtss パッケージで使用可能な tcsd(8) のマニュアルページも参照してください。

TCG ソフトウェアスタック (TSS) 仕様:
https://www.trustedcomputinggroup.org/specs/TSS (公開時点のアドレス)

注意事項

tpmadm は、tcsd サービス経由で TPM デバイスと通信します。tpmadm コマンドを使用する前に tcsd が動作している必要があります。tcsd が動作していない場合、tpmadm から次のエラーが生成されます。

```
Connect context: Communication failure (0x3011)
```

詳細は、tcsd(8) を参照してください。

名前

traceroute – print the route packets take to network host

形式

```
traceroute [-adFIlnSvx] [-A addr_family] [-c traffic_class]
     [-f first_hop] [-g gateway [-g gateway...] | -r]
     [-i iface] [-L flow_label] [-m max_hop]
     [-P pause_sec] [-p port] [-Q max_timeout]
     [-q nqueries] [-s src_addr] [-t tos] [-w wait_time] host
     [packetlen]
```

機能説明

The Internet is a large and complex aggregation of network hardware, connected by gateways. Tracking the route a packet follows can be difficult. The utility `traceroute` traces the route that an IP packet follows to another internet host.

The `traceroute` utility utilizes the both the IPv4 and IPv6 protocols. Use the `-A` option to override the default behavior. `traceroute` uses the IPv4 protocol *ttl* (time to live) field or the IPv6 field *hop limit*. It attempts to elicit an ICMP or ICMP6 `TIME_EXCEEDED` response from each *gateway* along the path, and a `PORT_UNREACHABLE`(or `ECHO_REPLY` if `-I` is used) response from the destination host. It starts by sending probes with a *ttl* or *hop limit* of `1` and increases by `1` until it either gets to the host, or it hits the maximum *max_hop*. The default maximum *max_hop* is 30 hops, but this can be set by the `-m` option.

Three probes are sent at each *ttl* (*hop limit*) setting, and a line is printed showing the *ttl* (*hop limit*), the hostname and the address of the gateway, and the *rtt* (round trip time) of each probe. The number of probes may be specifically set using the `-q` option. If the probe answers come from different gateways, the hostname and the address of each responding system will be printed. If there is no response within a 5 second timeout interval, an asterisk (`*`) is displayed for that probe. The `-w` option may be used to set the timeout interval. Other possible annotations that may appear after the time are:

`!`
: the *ttl* (*hop limit*) value in the received packet is `<= 1`.

`!H`
: host unreachable.

`!X`
: communication administratively prohibited.

`<!N>`
: ICMP (ICMP6) unreachable code N.

The following annotations appear only for IPv4:

`!F`
: fragmentation needed. This should never occur. If this is seen, the associated gateway is broken.

`!N`
: network unreachable.

`!P`
protocol unreachable.

`!S`
source route failed. It is likely that the gateway does not support source routing.

`!T`
unreachable for the specified tos (type-of-service).

`!U`
source host isolated or precedence problem.

The following annotations appear only for IPv6:

`!A`
host unreachable for a reason other than lack of an entry in the routing table.

`!B`
packet too big.

`!E`
destination is not a neighbor.

`!R`
unrecognized next header.

If almost all the probes result in some kind of unreachable code, then `traceroute` gives up and exits.

The destination *host* is not supposed to process the UDP probe packets, so the destination *port* default is set to an unlikely value. However, if some application on the destination is using that value, the value of *port* can be changed with the `-p` option.

The only mandatory parameter is the destination *host* name or IP number. The default probe datagram length is 40 bytes (60 bytes for IPv6), but this may be increased by specifying a packet length (in bytes) after the destination *host* name.

All integer arguments to `traceroute` can be specified in either decimal or hexadecimal notation. For example, *packetlen* can be specified either as `256` or `0x100`.

オプション

`-A` *addr_family*
Specify the address family of the target host. *addr_family* can be either `inet` or `inet6`. Address family determines which protocol to use. For an argument of `inet`, IPv4 is used. For `inet6`, IPv6 is used.

By default, if the name of a host is provided, not the literal IP address, and a valid IPv6 address exists in the name service database, `traceroute` will use this address. Otherwise, if the name service database contains an IPv4 address, it will try the IPv4 address.

Specify the address family `inet` or `inet6` to override the default behavior. If the argument specified is `inet`, `traceroute` will use the IPv4 address associated with the hostname. If none exists, `traceroute` will state that the host is unknown and exit. It will not try to determine if an IPv6 address exists in the name service database.

If the specified argument is `inet6`, `traceroute` will use the IPv6 address that is associated with the hostname. If none exists, `traceroute` will state that the host is unknown and exit.

`-a`
Probe all of the addresses of a multi-homed destination. The output looks like `traceroute` has been run once for each IP address of the destination. If this option is used together with `-A`, `traceroute` probes only the addresses that are of the specified address family. While probing one of the addresses of the destination, user can skip to the next address by sending a `SIGINT`, or exit `traceroute` by sending a `SIGQUIT` signal. See `signal(3C)`

`-c` *traffic_class*
Specify the traffic class of probe packets. The value must be an integer in the range from 0 to 255. Gateways along the path may route the probe packet differently depending upon the value of *traffic_class* set in the probe packet. This option is valid only on IPv6.

`-d`
Set the `SO_DEBUG` socket option.

`-F`
Turn off fragmentation. For IPv4, this means setting the Don't Fragment bit. For IPv4 and IPv6, this means do not allow fragmentation as the datagrams are sent. If the packetlen exceeds the MTU, then traceroute may report that sending failed due to Message too long.

`-f` *first_hop*
Set the starting *ttl* (*hop limit*) value to *first_hop*, to override the default value `1`. `traceroute` skips processing for those intermediate gateways which are less than *first_hop* hops away.

`-g` *gateway*
Specify a loose source route *gateway*. The user can specify more than one *gateway* by using `-g` for each gateway. The maximum number of gateways is 8 for IPv4 and 127 for IPv6. Note that some factors such as the link MTU can further limit the number of gateways for IPv6. This option cannot be used with the `-r` option.

Only users with the `{PRIV_NET_RAWACCESS}` privilege can specify a loose source route with this option.

`-I`
Use ICMP (ICMP6) `ECHO` instead of UDP datagrams.

`-i` *iface*
For IPv4, this option specifies a network interface to obtain the source IP address. This is normally only useful on a multi-homed host. The `-s` option is also another way to do this.

For IPv6, it specifies the network interface on which probe packets are transmitted. The argument can be either an interface index, for example, `1`, `2`, or an interface name, for example, `eri0`, `hme0`.

`-L` *flow_label*
Specify the flow label of probe packets. The value must be an integer in the range from 0 to 1048575. This option is valid only on IPv6.

`-l`
Print the value of the *ttl* (*hop limit*) field in each packet received.

`-m` *max_hop*
Set the maximum *ttl* (*hop limit*) used in outgoing probe packets. The default is 30 hops, which is the same default used for TCP connections.

`-n`
Print hop addresses numerically rather than symbolically and numerically. This saves a nameserver address-to-name lookup for each gateway found on the path.

`-P` *pause_sec*
Specify a delay, in seconds, to pause between probe packets. This may be necessary if the final destination does not accept undeliverable packets in bursts. By default, `traceroute` sends the next probe as soon as it has received a reply. Note that *pause_sec* is a real number.

`-p` *port*
Set the base UDP *port* number used in probes.The default is 33434. `traceroute` hopes that nothing is listening on UDP *ports* `(base+(nhops-1)*nqueries)` to `(base+(nhops*nqueries)-1)` at the destination host, so that an ICMP (ICMP6) `PORT_UNREACHABLE` message will be returned to terminate the route tracing. If something is listening on a *port* in the default range, this option can be used to select an unused *port* range. `nhops` is defined as the number of hops between the source and the destination.

`-Q` *max_timeout*
Stop probing this hop after *max_timeout* consecutive timeouts are detected. The default value is 5. Useful in combination with the `-q` option if you have specified a large *nqueries* probe count.

`-q` *nqueries*
Set the desired number of probe queries. The default is 3.

`-r`
Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to send probes to a local host through an interface that has been dropped by the router daemon. See `in.routed(1M)`. You cannot use this option if the `-g` option is used.

`-S`
Display a summary of how many probes were not answered for each hop.

`-s` *src_addr*
: Use the following address, which usually is given as a literal IP address, not a hostname, as the source address in outgoing probe packets. On multi-homed hosts, those with more than one IP address, this option can be used to force the source address to be something other than the IP address `traceroute` picks by default. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent. For IPv4, when used together with the `-i` option, the given IP address should be configured on the specified interface. Otherwise, an error will be returned. In the case of IPv6, the interface name and the source address do not have to match.

`-t` *tos*
: Set the *tos*(type-of-service) in probe packets to the specified value. The default is zero. The value must be an integer in the range from 0 to 255. Gateways along the path may route the probe packet differently depending upon the *tos* value set in the probe packet. This option is valid only on IPv4.

`-v`
: Verbose output. For each hop, the size and the destination of the response packets is displayed. Also ICMP (ICMP6) packets received other than `TIME_EXCEEDED` and `UNREACHABLE` are listed as well.

`-w` *waittime*
: Set the time, in seconds, to wait for a response to a probe. The default is 5 seconds.

`-x`
: Prevent `traceroute` from calculating checksums. Checksums are usually required for the last hop when using ICMP `ECHO` probes. This option is valid only on IPv4. See the `-I` option.

  When specified from within a shared-IP zone, this option has no effect as the checksum is always calculated by the operating system in this case.

オペランド

The following operands are supported:

host
: The network host.

使用例

例 1　Sample Output From the traceroute Utility

Some sample output from the `traceroute` utility might be:

```
istanbul% traceroute london
traceroute: Warning: london has multiple addresses;  \
  using 4::114:a00:20ff:ab3d:83ed
traceroute: Warning: Multiple interfaces found; \
  using 4::56:a00:20ff:fe93:8dde @ eri0:2
traceroute to london (4::114:a00:20ff:ab3d:83ed), 30 hops max, \
  60 byte packets
1  frbldg7c-86 (4::56:a00:20ff:fe1f:65a1)  1.786 ms  1.544 ms  1.719 ms
```

例 1 Sample Output From the traceroute Utility *(続き)*

```
2  frbldg7b-77 (4::255:0:0:c0a8:517)  2.587 ms 3.001 ms  2.988 ms
3  london (4::114:a00:20ff:ab3d:83ed)  3.122 ms  2.744 ms  3.356 ms
```

The target host, london, has both IPv4 and IPv6 addresses in the name service database. According to the default behavior, `traceroute` uses IPv6 address of the destination host.

例 2 Using the traceroute Utility For a Host Which has Only IPv4 Addresses

In the following examples, `traceroute` is tracking the route to host sanfrancisco, which has only IPv4 addresses in the name service database. Therefore traceroute uses only IPv4 addresses. The following shows the 7-hop path that a packet would follow from the host istanbul to the host sanfrancisco.

```
istanbul% traceroute sanfrancisco
traceroute: Warning: Multiple interfaces found; using 172.31.86.247 @eri0
traceroute to sanfrancisco (172.29.64.39), 30 hops max, 40 byte packets
1  frbldg7c-86 (172.31.86.1)  1.516 ms  1.283 ms  1.362 ms
2  bldg1a-001 (172.31.1.211)  2.277 ms  1.773 ms  2.186 ms
3  bldg4-bldg1 (172.30.4.42)  1.978 ms  1.986 ms  13.996 ms
4  bldg6-bldg4 (172.30.4.49)  2.655 ms  3.042 ms  2.344 ms
5  ferbldg11a-001 (172.29.1.236)  2.636 ms  3.432 ms  3.830 ms
6  frbldg12b-153 (172.29.153.72)  3.452 ms  3.146 ms  2.962 ms
7  sanfrancisco (172.29.64.39)  3.430 ms  3.312 ms  3.451 ms
```

例 3 Using the traceroute Utility With Source Routing

The following example shows the path of a packet that goes from istanbul to sanfrancisco through the hosts cairo and paris, as specified by the `-g` option. The `-I` option makes `traceroute` send ICMP `ECHO` probes to the host sanfrancisco. The `-i` options sets the source address to the IP address configured on the interface `qe0`.

```
istanbul% traceroute -g cairo -g paris -i qe0 -q 1 -I sanfrancisco
traceroute to sanfrancisco (172.29.64.39), 30 hops max, 56 byte packets
1  frbldg7c-86 (172.31.86.1)  2.012 ms
2  flrbldg7u (172.31.17.131)  4.960 ms
3  cairo (192.168.163.175) 4.894 ms
4  flrbldg7u (172.31.17.131)  3.475 ms
5  frbldg7c-017 (172.31.17.83)  4.126 ms
6  paris (172.31.86.31) 4.086 ms
7  frbldg7b-82 (172.31.82.1)  6.454 ms
8  bldg1a-001 (172.31.1.211)  6.541 ms
9  bldg6-bldg4 (172.30.4.49)  6.518 ms
10  ferbldg11a-001 (172.29.1.236)  9.108 ms
11  frbldg12b-153 (172.29.153.72)  9.634 ms
12  sanfrancisco (172.29.64.39)  14.631 ms
```

終了ステータス

The following exit values are returned:

`0`
Successful operation.

`>0`
An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`netstat(1M)`, `signal(3C)`, `ping(1M)`, `attributes(5)`, `privileges(5)`, `zones(5)`

警告

This utility is intended for use in network testing, measurement and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use `traceroute`(1M) during normal operations or from automated scripts.

名前

trapstat – report trap statistics

形式

```
/usr/sbin/trapstat [-t | -T | -e entry]
     [-C processor_set_id | -c cpulist] [-P] [-a]
     [-r rate] [ [interval [count]] | command | [args]]

/usr/sbin/trapstat -l
```

機能説明

The `trapstat` utility gathers and displays run-time trap statistics on UltraSPARC-based systems. The default output is a table of trap types and CPU IDs, with each row of the table denoting a trap type and each column of the table denoting a CPU. If standard output is a terminal, the table contains as many columns of data as can fit within the terminal width; if standard output is not a terminal, the table contains at most six columns of data. By default, data is gathered and and displayed for all CPUs; if the data cannot fit in a single table, it is printed across multiple tables. The set of CPUs for which data is gathered and displayed can be optionally specified with the `-c` or `-C` option.

Unless the `-r` option or the `-a` option is specified, the value displayed in each entry of the table corresponds to the number of traps per second. If the `-r` option is specified, the value corresponds to the number of traps over the interval implied by the specified sampling rate; if the `-a` option is specified, the value corresponds to the accumulated number of traps since the invocation of `trapstat`.

By default, `trapstat` displays data once per second, and runs indefinitely; both of these behaviors can be optionally controlled with the *interval* and *count* parameters, respectively. The *interval* is specified in seconds; the *count* indicates the number of intervals to be executed before exiting. Alternatively, `command` can be specified, in which case `trapstat` executes the provided command and continues to run until the command exits. A positive integer is assumed to be an *interval*; if the desired *command* cannot be distinguished from an integer, the full path of *command* must be specified.

UltraSPARC I (obsolete), II, and III handle translation lookaside buffer (TLB) misses by trapping to the operating system. TLB miss traps can be a significant component of overall system performance for some workloads; the `-t` option provides in-depth information on these traps. When run with this option, `trapstat` displays both the rate of TLB miss traps and the percentage of time spent processing those traps. Additionally, TLB misses that hit in the translation storage buffer (TSB) are differentiated from TLB misses that further miss in the TSB. (The TSB is a software structure used as a translation entry cache to allow the TLB to be quickly filled; it is discussed in detail in the *UltraSPARC II User's Manual*.) The TLB and TSB miss information is further broken down into user- and kernel-mode misses.

Workloads with working sets that exceed the TLB reach may spend a significant amount of time missing in the TLB. To accommodate such workloads, the operating system supports multiple page sizes: larger page sizes increase the effective TLB reach and thereby reduce the number of TLB misses. To provide insight into the relationship between page size and TLB miss rate, `trapstat` optionally provides in-depth TLB miss information broken down by page size using the `-T` option. The information provided by the `-T` option is a superset of that provided by the `-t` option; only one of `-t` and `-T` can be specified.

オプション

The following options are supported:

`-a`

Displays the number of traps as accumulating, monotonically increasing values instead of per-second or per-interval rates.

`-c` *cpulist*

Enables `trapstat` only on the CPUs specified by *cpulist*.

cpulist can be a single processor ID (for example, `4`), a range of processor IDs (for example, `4-6`), or a comma separated list of processor IDs or processor ID ranges (for example, `4,5,6` or `4,6-8`).

`-C` *processor_set_id*

Enables `trapstat` only on the CPUs in the processor set specified by *processor_set_id*.

`trapstat` modifies its output to always reflect the CPUs in the specified processor set. If a CPU is added to the set, `trapstat` modifies its output to include the added CPU; if a CPU is removed from the set, `trapstat` modifies its output to exclude the removed CPU. At most one processor set can be specified.

`-e` *entrylist*

Enables `trapstat` only for the trap table entry or entries specified by *entrylist*. A trap table entry can be specified by trap number or by trap name (for example, the level–10 trap can be specified as `74`, `0x4A`, `0x4a`, or `level-10`).

entrylist can be a single trap table entry or a comma separated list of trap table entries. If the specified trap table entry is not valid, `trapstat` prints a table of all valid trap table entries and values. A list of valid trap table entries is also found in *The SPARC Architecture Manual, Version 9* and the *Sun Microelectronics UltraSPARC II User's Manual*. If the parsable option (`-P`) is specified in addition to the `-e` option, the format of the data is as follows:

Field	Contents
1	Timestamp (nanoseconds since start)
2	CPU ID
3	Trap number (in hexadecimal)
4	Trap name
5	Trap rate per interval

Each field is separated with whitespace. If the format is modified, it will be modified by adding potentially new fields beginning with field 6; exant fields will remain unchanged.

`-l`

Lists trap table entries. By default, a table is displayed containing all valid trap numbers, their names and a brief description. The trap name is used in both the default output and in the *entrylist* parameter for the `-e` argument. If the parsable option (`-P`) is specified in addition to the `-l` option, the format of the data is as follows:

Field	Contents
1	Trap number in hexadecimal
2	Trap number in decimal
3	Trap name
Remaining	Trap description

`-P`

Generates parsable output. When run without other data gathering modifying options (that is, `-e`, `-t` or `-T`), `trapstat`'s the parsable output has the following format:

Field	Contents
1	Timestamp (nanoseconds since start)
2	CPU ID
3	Trap number (in hexadecimal)
4	Trap name
5	Trap rate per interval

Each field is separated with whitespace. If the format is modified, it will be modified by adding potentially new fields beginning with field 6; extant fields will remain unchanged.

`-r` *rate*

Explicitly sets the sampling rate to be *rate* samples per second. If this option is specified, `trapstat`'s output changes from a traps-per-second to traps-per-sampling-interval.

`-t`

Enables TLB statistics.

A table is displayed with four principal columns of data: *itlb-miss*, *itsb-miss*, *dtlb-miss*, and *dtsb-miss*. The columns contain both the rate of the corresponding event and the percentage of CPU time spent processing the event. The percentage of CPU time is given only in terms of a single CPU. The rows of the table correspond to CPUs,

with each CPU consuming two rows: one row for user-mode events (denoted with u) and one row for kernel-mode events (denoted with k). For each row, the percentage of CPU time is totalled and displayed in the rightmost column. The CPUs are delineated with a solid line. If the parsable option (-P) is specified in addition to the -t option, the format of the data is as follows:

Field	Contents
1	Timestamp (nanoseconds since start)
2	CPU ID
3	Mode (k denotes kernel, u denotes user)
4	I-TLB misses
5	Percentage of time in I-TLB miss handler
6	I-TSB misses
7	Percentage of time in I-TSB miss handler
8	D-TLB misses
9	Percentage of time in D-TLB miss handler
10	D-TSB misses
11	Percentage of time in D-TSB miss handler

Each field is separated with whitespace. If the format is modified, it will be modified by adding potentially new fields beginning with field 12; extant fields will remain unchanged.

-T

Enables TLB statistics, with page size information. As with the -t option, a table is displayed with four principal columns of data: *itlb-miss*, *itsb-miss*, *dtlb-miss*, and *dtsb-miss*. The columns contain both the absolute number of the corresponding event, and the percentage of CPU time spent processing the event. The percentage of CPU time is given only in terms of a single CPU. The rows of the table correspond to CPUs, with each CPU consuming two sets of rows: one set for user-level events (denoted with u) and one set for kernel-level events (denoted with k). Each set, in turn, contains as many rows as

there are page sizes supported (see getpagesizes(3C)). For each row, the percentage of CPU time is totalled and displayed in the right-most column. The two sets are delineated with a dashed line; CPUs are delineated with a solid line. If the parsable option (-P) is specified in addition to the -T option, the format of the data is as follows:

Field	Contents
1	Timestamp (nanoseconds since start)
2	CPU ID
3	Mode k denotes kernel, u denotes user)
4	Page size, in decimal
5	I-TLB misses
6	Percentage of time in I-TLB miss handler
7	I-TSB misses
8	Percentage of time in I-TSB miss handler
9	D-TLB misses
10	Percentage of time in D-TLB miss handler
11	D-TSB misses
12	Percentage of time in D-TSB miss handler

Each field is separated with whitespace. If the format is modified, it will be modified by adding potentially new fields beginning with field 13; extant fields will remain unchanged.

使用例

例 1 Using trapstat Without Options

When run without options, trapstat displays a table of trap types and CPUs. At most six columns can fit in the default terminal width; if (as in this example) there are more than six CPUs, multiple tables are displayed:

```
example# trapstat
vct  name                |     cpu0     cpu1     cpu4     cpu5     cpu8     cpu9
-------------------------+-------------------------------------------------------
 24 cleanwin             |     6446     4837     6368     2153     2623     1321
 41 level-1              |      100        0        0        0        1        0
```

例 1　Using trapstat Without Options　　*(続き)*

```
 44 level-4             |        0        1        1        1        0        0
 45 level-5             |        0        0        0        0        0        0
 47 level-7             |        0        0        0        0        9        0
 49 level-9             |      100      100      100      100      100      100
 4a level-10            |      100        0        0        0        0        0
 4d level-13            |        6       10        7       16       13       11
 4e level-14            |      100        0        0        0        1        0
 60 int-vec             |     2607     2740     2642     2922     2920     3033
 64 itlb-miss           |     3129     2475     3167     1037     1200      569
 68 dtlb-miss           |   121061    86162   109838    37386    45639    20269
 6c dtlb-prot           |      997      847     1061      379      406      184
 84 spill-user-32       |     2809     2133     2739   200806   332776   454504
 88 spill-user-64       |    45819   207856    93487   228529    68373    77590
 8c spill-user-32-cln   |      784      561      767      274      353      215
 90 spill-user-64-cln   |        9       37       17       39       12       13
 98 spill-kern-64       |    62913    50145    63869    21916    28431    11738
 a4 spill-asuser-32     |     1327      947     1288      460      572      335
 a8 spill-asuser-64     |       26       48       18       54       10       14
 ac spill-asuser-32-cln |     4580     3599     4555     1538     1978      857
 b0 spill-asuser-64-cln |       26        0        0        2        0        0
 c4 fill-user-32        |     2862     2161     2798   191746   318115   435850
 c8 fill-user-64        |    45813   197781    89179   217668    63905    74281
 cc fill-user-32-cln    |     3802     2833     3733    10153    16419    19475
 d0 fill-user-64-cln    |      329    10105     4873    10603     4235     3649
 d8 fill-kern-64        |    62519    49943    63611    21824    28328    11693
108 syscall-32          |     2285     1634     2278      737      957      383
126 self-xcall          |      100        0        0        0        0        0

vct  name               |    cpu12    cpu13    cpu14    cpu15
------------------------+-----------------------------------
 24 cleanwin            |     5435     4232     6302     6104
 41 level-1             |        0        0        0        0
 44 level-4             |        2        0        0        1
 45 level-5             |        0        0        0        0
 47 level-7             |        0        0        0        0
 49 level-9             |      100      100      100      100
 4a level-10            |        0        0        0        0
 4d level-13            |       15       11       22       11
 4e level-14            |        0        0        0        0
 60 int-vec             |     2813     2833     2738     2714
 64 itlb-miss           |     2636     1925     3133     3029
 68 dtlb-miss           |    90528    70639   107786   103425
 6c dtlb-prot           |      819      675      988      954
 84 spill-user-32       |   175768    39933     2811     2742
 88 spill-user-64       |        0   241348    96907   118298
 8c spill-user-32-cln   |      681      513      753      730
```

例 1 Using trapstat Without Options (続き)

```
 90 spill-user-64-cln   |        0       42       16       20
 98 spill-kern-64       |    52158    40914    62305    60141
 a4 spill-asuser-32     |     1113      856     1251     1208
 a8 spill-asuser-64     |        0       64       16       24
 ac spill-asuser-32-cln |     3816     2942     4515     4381
 b0 spill-asuser-64-cln |        0        0        0        0
 c4 fill-user-32        |   170744    38444     2876     2784
 c8 fill-user-64        |        0   230381    92941   111694
 cc fill-user-32-cln    |     8550     3790     3612     3553
 d0 fill-user-64-cln    |        0    10726     4495     5845
 d8 fill-kern-64        |    51968    40760    62053    59922
108 syscall-32          |     1839     1495     2144     2083
126 self-xcall          |        0        0        0        0
```

例 2 Using trapset with CPU Filtering

The `-c` option can be used to limit the CPUs on which `trapstat` is enabled. This example limits `CPU 1` and CPUs 12 through 15.

```
example# trapstat -c 1,12-15
```

```
vct  name               |     cpu1    cpu12    cpu13    cpu14    cpu15
------------------------+----------------------------------------------
 24 cleanwin            |     6923     3072     2500     3518     2261
 44 level-4             |        3        0        0        1        1
 49 level-9             |      100      100      100      100      100
 4d level-13            |       23        8       14       19       14
 60 int-vec             |     2559     2699     2752     2688     2792
 64 itlb-miss           |     3296     1548     1174     1698     1087
 68 dtlb-miss           |   114788    54313    43040    58336    38057
 6c dtlb-prot           |     1046      549      417      545      370
 84 spill-user-32       |    66551    29480   301588    26522   213032
 88 spill-user-64       |        0   318652   111239   299829   221716
 8c spill-user-32-cln   |      856      347      331      416      293
 90 spill-user-64-cln   |        0       55       21       59       39
 98 spill-kern-64       |    66464    31803    24758    34004    22277
 a4 spill-asuser-32     |     1423      569      560      698      483
 a8 spill-asuser-64     |        0       74       32       98       46
 ac spill-asuser-32-cln |     4875     2250     1728     2384     1584
 b0 spill-asuser-64-cln |        0        2        0        1        0
 c4 fill-user-32        |    64193    28418   287516    27055   202093
 c8 fill-user-64        |        0   305016   106692   288542   210654
 cc fill-user-32-cln    |     6733     3520    15185     2396    12035
 d0 fill-user-64-cln    |        0    13226     3506    12933    11032
 d8 fill-kern-64        |    66220    31680    24674    33892    22196
108 syscall-32          |     2446      967      817     1196      755
```

例 3 Using trapstat with TLB Statistics

The -t option displays in-depth TLB statistics, including the amount of time spent performing TLB miss processing. The following example shows that the machine is spending 14.1 percent of its time just handling D-TLB misses:

```
example# trapstat -t
cpu m| itlb-miss %tim itsb-miss %tim | dtlb-miss %tim dtsb-miss %tim |%tim
-----+-------------------------------+-------------------------------+----
  0 u|      2571  0.3         0  0.0 |     10802  1.3         0  0.0 | 1.6
  0 k|         0  0.0         0  0.0 |    106420 13.4       184  0.1 |13.6
-----+-------------------------------+-------------------------------+----
  1 u|      3069  0.3         0  0.0 |     10983  1.2       100  0.0 | 1.6
  1 k|        27  0.0         0  0.0 |    106974 12.6        19  0.0 |12.7
-----+-------------------------------+-------------------------------+----
  2 u|      3033  0.3         0  0.0 |     11045  1.2       105  0.0 | 1.6
  2 k|        43  0.0         0  0.0 |    107842 12.7       108  0.0 |12.8
-----+-------------------------------+-------------------------------+----
  3 u|      2924  0.3         0  0.0 |     10380  1.2       121  0.0 | 1.6
  3 k|        54  0.0         0  0.0 |    102682 12.2        16  0.0 |12.2
-----+-------------------------------+-------------------------------+----
  4 u|      3064  0.3         0  0.0 |     10832  1.2       120  0.0 | 1.6
  4 k|        31  0.0         0  0.0 |    107977 13.0       236  0.1 |13.1
=====+===============================+===============================+====
 ttl |     14816  0.3         0  0.0 |    585937 14.1      1009  0.0 |14.5
```

例 4 Using trapstat with TLB Statistics and Page Size Information

By specifying the -T option, trapstat shows TLB misses broken down by page size. In this example, CPU 0 is spending 7.9 percent of its time handling user-mode TLB misses on 8K pages, and another 2.3 percent of its time handling user-mode TLB misses on 64K pages.

```
example# trapstat -T -c 0
cpu m size| itlb-miss %tim itsb-miss %tim | dtlb-miss %tim dtsb-miss %tim |%tim
----------+-------------------------------+-------------------------------+----
  0 u   8k|      1300  0.1        15  0.0 |    104897  7.9        90  0.0 | 8.0
  0 u  64k|         0  0.0         0  0.0 |     29935  2.3         7  0.0 | 2.3
  0 u 512k|         0  0.0         0  0.0 |      3569  0.2         2  0.0 | 0.2
  0 u   4m|         0  0.0         0  0.0 |       233  0.0         2  0.0 | 0.0
- - - - - + - - - - - - - - - - - - - - - + - - - - - - - - - - - - - - - + - -
  0 k   8k|        13  0.0         0  0.0 |     71733  6.5       110  0.0 | 6.5
  0 k  64k|         0  0.0         0  0.0 |         0  0.0         0  0.0 | 0.0
  0 k 512k|         0  0.0         0  0.0 |         0  0.0       206  0.1 | 0.1
  0 k   4m|         0  0.0         0  0.0 |         0  0.0         0  0.0 | 0.0
==========+===============================+===============================+====
      ttl |      1313  0.1        15  0.0 |    210367 17.1       417  0.2 |17.5
```

例 5 Using trapstat with Entry Filtering

By specifying the `-e` option, `trapstat` displays statistics for only specific trap types. Using this option minimizes the probe effect when seeking specific data. This example yields statistics for only the *dtlb-prot* and *syscall-32* traps on CPUs 12 through 15:

```
example# trapstat -e dtlb-prot,syscall-32 -c 12-15
vct  name                |    cpu12    cpu13    cpu14    cpu15
------------------------+-----------------------------------
 6c dtlb-prot            |      817      754     1018      560
108 syscall-32           |     1426     1647     2186     1142

vct  name                |    cpu12    cpu13    cpu14    cpu15
------------------------+-----------------------------------
 6c dtlb-prot            |     1085      996      800      707
108 syscall-32           |     2578     2167     1638     1452
```

例 6 Using trapstat with a Higher Sampling Rate

The following example uses the `-r` option to specify a sampling rate of 1000 samples per second, and filter only for the level-10 trap. Additionally, specifying the `-P` option yields parsable output.

Notice the timestamp difference between the level-10 events: 9,998,000 nanoseconds and 10,007,000 nanoseconds. These level-10 events correspond to the system clock, which by default ticks at 100 hertz (that is, every 10,000,000 nanoseconds).

```
example# trapstat -e level-10 -P -r 1000
1070400 0 4a level-10 0
2048600 0 4a level-10 0
3030400 0 4a level-10 1
4035800 0 4a level-10 0
5027200 0 4a level-10 0
6027200 0 4a level-10 0
7027400 0 4a level-10 0
8028200 0 4a level-10 0
9026400 0 4a level-10 0
10029600 0 4a level-10 0
11028600 0 4a level-10 0
12024000 0 4a level-10 0
13028400 0 4a level-10 1
14031200 0 4a level-10 0
15027200 0 4a level-10 0
16027600 0 4a level-10 0
17025000 0 4a level-10 0
18026000 0 4a level-10 0
19027800 0 4a level-10 0
20025600 0 4a level-10 0
21025200 0 4a level-10 0
22025000 0 4a level-10 0
```

例 6 Using trapstat with a Higher Sampling Rate (続き)

```
23035400 0 4a level-10 1
24027400 0 4a level-10 0
25026000 0 4a level-10 0
26027000 0 4a level-10 0
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	
Human Readable Output	Uncommitted
Parsable Output	Committed

関連項目

lockstat(1M), pmap(1), psrset(1M), psrinfo(1M), pbind(1M), ppgsz(1), getpagesizes(3C)

Sun Microelectronics UltraSPARC II User's Manual, January 1997, STP1031,

The SPARC Architecture Manual, Version 9, 1994, Prentice-Hall.

注意事項

When enabled, trapstat induces a varying probe effect, depending on the type of information collected. While the precise probe effect depends upon the specifics of the hardware, the following table can be used as a rough guide:

Option	Approximate probe effect
default	3-5% per trap
-e	3-5% per specified trap
-t, -T	40-45% per TLB miss trap hitting in the TSB, 25-30% per TLB miss trap missing in the TSB

These probe effects are *per trap* not for the system as a whole. For example, running trapstat with the default options on a system that spends 7% of total time handling traps induces a performance degradation of less than one half of one percent; running trapstat with the -t or -T option on a system spending 5% of total time processing TLB misses induce a performance degradation of no more than 2.5%.

When run with the -t or -T option, trapstat accounts for its probe effect when calculating the *%tim* fields. This assures that the *%tim* fields are a reasonably accurate indicator of the time a given workload is spending handling TLB misses — regardless of the perturbing presence of trapstat.

While the *%tim* fields include the explicit cost of executing the TLB miss handler, they do *not* include the implicit costs of TLB miss traps (for example, pipeline effects, cache pollution, etc). These implicit costs become more significant as the trap rate grows; if high *%tim* values are reported (greater than 50%), you can accurately infer that much of the balance of time is being spent on the implicit costs of the TLB miss traps.

Due to the potential system wide degradation induced, only the super-user can run `trapstat`.

Due to the limitation of the underlying statistics gathering methodology, only one instance of `trapstat` can run at a time.

名前

ttymon – 端末ポートのポートモニター

形式

```
/usr/lib/ttymon -g [-d device] [-h] [-t timeout]
      [-l ttylabel] [-p prompt] [-m modules] [-T termtype]
```

機能説明

ttymon は STREAMS ベースの TTY ポートモニターです。その機能には、ポートの監視、ポートの端末モード、ボーレートおよび回線規約の設定、そしてポートに関連したサービスへのユーザーまたはアプリケーションの接続があります。ttymon の各インスタンスは、起動時に指定された 1 つのポートを監視します。ttymon のインスタンスが起動されると、ttymon は最初に回線制御手順 (指定されている場合)、および速度と端末の設定を初期化します。/etc/logindevperm 中のエントリに指定されているポートには、デバイスのオーナー、グループ、アクセス権が設定されます (logindevperm(4) 参照)。初期化に使用される値は、TTY 設定ファイルの適切なエントリから取り込まれます。このファイルは sttydefs(1M) コマンドによって管理されています。通常、ポート上のデフォルトの回線規約は、オートプッシュファシリティの autopush(1M) コマンドによって設定されます。

次に ttymon はプロンプトを書き込んでユーザーの入力を待ちます。ユーザーが BREAK キーを押して速度が適切でないことを知らせると、ttymon は次の速度を試行してプロンプトを再び書き込みます。有効な入力が受信されると、ttymon は utmpx エントリ (utmpx(4) を参照) を作成し、ポートのログインサービスを実行します。有効な入力は、キャリッジリターンで終わり、少なくとも 1 つの復帰改行以外の文字が入った文字列からなります。

autobaud がポートで動作可能であれば、ttymonは自動的にポートのボーレートを判別しようとします。ユーザーがキャリッジリターンを入力してからでないと、ttymonはボーレートを認識してプロンプトをプリントできません。現在のところ、*autobaud* が判別できるボーレートは 110、1200、2400、4800、および 9600 です。

SMF サービス記述

ttymon を起動するプライマリ smf(5) サービスは、svc:/system/console-login です。このサービスには、複数のサービスインスタンスが含まれる場合があります。インスタンスについては、以下でより詳しく説明します。次に示すように、このサービスは呼び出しを制御するためにプロパティーグループ ttymon 内の多くのプロパティーを提供します。

NAME	TYPE	TTYMON OPTION
device	astring	[-d device]
nohangup	boolean	[-h]
label	astring	[-l label]
modules	astring	[-m module1,module2]
prompt	astring	[-p prompt]
timeout	count	[-t timeout]
terminal_type	astring	[-T termtype]

いずれかの値が空の文字列であるかゼロに設定された整数である場合、オプションは ttymon 呼び出しに渡されません。

`svc:/system/console-login:default`
: デフォルトインスタンスは常に、システムハードウェアコンソールでのログインを提供する ttymon を表します。

  システムコンソール設定の変更方法の例については、「使用例」を参照してください。

`svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`
: システムの仮想コンソールには、追加のサービスインスタンスが提供されています。 仮想コンソールが使用できない場合は、これらのサービスが自動的に無効化されます。 vtdaemon(1M) を参照してください。

`svc:/system/console-login:{terma, termb}`
: 便宜のために svc:/system/console-login:terma および svc:/system/console-login:termb が提供されており、追加ポート /dev/term/a および /dev/term/b にログインサービスを設定する際にユーザーが使用できます。これらのサービスはデフォルトで無効化されています。

追加インスタンスを作成する

ユーザーは追加デバイスに追加のサービスインスタンスを構成できます。これは、次の方法のいずれかで実現できます。

- svccfg(1M) を使用して手動でサービスインスタンスを作成する。
- サービスプロファイルにサービスを作成する (smf(5) を参照)。
- 追加のサービスインスタンスにサービスマニフェストを作成する。

svccfg を使用して手動でサービスを構成する例については、「使用例」を参照してください。

SMF サービスのエラー

コンソールログインサービスのインスタンスが誤って構成された場合はほとんど、保守状態に遷移します。svcs -l を使用して (svcs(1) を参照)、サービスのログファイルの場所を確認し、追加情報がないかどうかログを調べます。

エラーが発生すると、サービスが無制限に再起動されることがあります。svcadm(1M) を使用してサービスを無効化してから、問題解決に役立つ追加のメッセージや情報がないかどうかサービスログを調べます。

セキュリティー

ttymon は pam(3PAM)を使って、セッション管理を行います。PAM 構成ポリシーは ttymon で使用するモジュールを明記しています。このポリシーは /etc/pam.conf で見ることができます。以下に UNIX セッション管理モジュールを使用する ttymon コマンドのエントリの入った pam.conf ファイルの抜粋を示します。

```
ttymon session required /usr/lib/security/pam_unix_session.so.1
```

ttymon サービスのエントリがない場合には「other」のサービスのエントリを使用します。

オプション

次のオプションがサポートされています。

- `-g` 歴史的経緯により、`-g` オプションは必須です。
- `-d`*device* *device* は `ttymon` の接続先とすべきポートの完全パス名です。このオプションが指定されていなければ、ファイル記述子 `0` は TTY ポートに対する呼び出しプロセスによって設定される必要があります。
- `-h` -h フラグが指定されていないと、`ttymon` は、速度をデフォルト速度または指定速度に設定する前に、速度を 0 に設定することによって回線のハングアップを強制的に行います。
- `-l`*ttylabel* *ttylabel* は `ttydefs` ファイルの速度および TTY 定義に対するリンクです。この定義によって、初期の実行速度、初期の TTY 設定の内容およびユーザーがBREAK キーを押して速度が適切でないことを指示する場合に、次に試行する速度が `ttymon` に通知されます。デフォルト速度は 9600 ボーです。
- `-m`*modules* ポートを初期化すると、`ttymon` はポートのモジュールすべてをポップして、指定した順序で *modules* をプッシュします。*modules* はコンマで区切ったプッシュ可能なモジュールのリストです。通常、ポートのデフォルトのモジュールは、オートプッシュファシリティによって設定されます。
- `-p`*prompt* ユーザーがプロンプト文字列を指定するのを許可します。デフォルトのプロンプトは `Login:` です。
- `-t`*timeout* プロンプトの送信後、*timeout* 秒内に何も入力がない場合には `ttymon` を終了します。
- `-T`*termtype* *termtype* に `TERM`環境変数を設定します。
- `-v` 詳細なメッセージを有効化します。

使用例

例 1 システムコンソールの端末タイプを設定する

次の例では、システムコンソール `ttymon` 呼び出しのための端末タイプ (`-T`) オプションの値を設定します。

```
# svccfg -s svc:/system/console-login:default \
        "setprop ttymon/terminal_type = xterm"
# svcadm refresh svc:/system/console-login:default
```

例 2 追加シリアルデバイスのサービスインスタンスを作成する

この例では、ユーザーが USB シリアルアダプタを使用して接続されている端末でログインサービスを提供するために、`svc:/system/console-login` サービスの追加インスタンスを構成します。USB シリアルポートが `/dev/term/1` として存在

例 2 追加シリアルデバイスのサービスインスタンスを作成する (続き)

し、ユーザーがこれに vt100 端末を接続する計画があると想定しています。 この場合、サービスインスタンスには term1 (またはその他の任意の名前) という名前を付け、次のように定義できます。

```
# svccfg -s svc:/system/console-login "add term1"
# SVC=svc:/system/console-login:term1
# svccfg -s $SVC "addpg ttymon application"
# svccfg -s $SVC "setprop ttymon/device = /dev/term/1"
# svccfg -s $SVC "setprop ttymon/terminal_type = vt100"
# svcadm refresh $SVC
# svcadm enable $SVC
```

環境変数

LC_* 変数
(LC_CTYPE、LC_MESSAGES、LC_TIME、LC_COLLATE、LC_NUMERIC、LC_MONETARY) (environ(5)参照) のいずれも環境に設定されていなければ、それぞれ対応するロケール のカテゴリにおける ttymon の動作は、環境変数 LANG によって決定されます。LC_ALL が設定されていれば、その内容が LANG 変数やその他の LC_* 変数より優先されます。前述の変数が環境にまったく設定されていなければ、「C」ロケール (米国スタイル) が ttymon の動作を決定します。

LC_CTYPE ttymon の文字の処理方法を決定します。LC_CTYPE に有効な値が設定されていると、ttymon は、そのロケールにあった文字を含むテキストやファイル名を表示および処理できます。ttymon は拡張 UNIX コード(EUC)も表示および処理できます。この場合、文字は 1 バイト幅、 2 バイト幅、3 バイト幅のいずれも使用できます。また、ttymon は 1、 2、またはそれ以上のカラム幅の EUC 文字も処理することができます。「C」ロケール においては、ISO 8859-1の文字だけが有効です。

ファイル

/etc/logindevperm
コンソールデバイスのログインまたはログアウト時にデバイスの所有者、グループ、およびアクセス権を変更するときに、login(1) および ttymon で使用される情報が含まれます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

svcs(1), ct(1C), cu(1C), autopush(1M), sttydefs(1M), svcadm(1M), svccfg(1M), uucico(1M), vtdaemon(1M), pam(3PAM), logindevperm(4), pam.conf(4), utmpx(4), attributes(5), environ(5), pam_authtok_check(5), pam_authtok_get(5),

pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5), smf(5)

『Oracle Solaris の管理: 一般的なタスク』

注意事項

サービスアクセス機能 (SAF と SAC)

以前は、ttymon はサービスアクセス機能のコンポーネントであり、sac (サービスアクセスコントローラ) によって起動されていました。このリリースの Solaris では、この機能は削除され、SMF への関連部分の変換が行われました。

ポートの競合

ポートが複数の ttymon によって監視されている場合は、ttymon は入力を争うような方法でプロンプトメッセージを送信できます。

2 つの svc:/system/console-login サービスインスタンスが同じ基本デバイスを参照している可能性があります。たとえば、(ファームウェアの設定や自動検出のために) システムのハードウェアコンソールがシリアルポート A に接続されている場合は、svc:/system/console-login:default と svc:/system/console-login:terma の両方のサービスが同じ基本ハードウェアデバイスを参照します。この状況を回避するために追加サービスインスタンスを定義または有効化する際は、注意しないと 2 つの ttymon が競合してしまいます。

名前

tunefs – tune an existing UFS file system

形式

```
tunefs [-a maxcontig] [-d rotdelay] [-e maxbpg]
     [-m minfree] [-o space | time] special | filesystem
```

機能説明

`tunefs` is designed to change the dynamic parameters of a file system that affect the layout policies. When using `tunefs` with *filesystem*, *filesystem* must be in `/etc/vfstab`. The parameters that can be changed are indicated by the options given below.

オプション

The following options are supported:

`-a` *maxcontig*

The maximum number of logical blocks, belonging to one file, that is allocated contiguously. The default is calculated as follows:

`maxcontig =` *disk drive maximum transfer size / disk block size*

If the disk drive's maximum transfer size cannot be determined, the default value for `maxcontig` is calculated from kernel parameters as follows:

If `maxphys` is less than `ufs_maxmaxphys`, which is 1 Mbyte, then `maxcontig` is set to `maxphys`. Otherwise, `maxcontig` is set to `ufs_maxmaxphys`.

You can set `maxcontig` to any positive integer value.

The actual value will be the lesser of what has been specified and what the hardware supports.

`-d` *rotdelay*

This parameter is obsolete as of the Solaris 10 release. The value is always set to `0`, regardless of the input value.

`-e` *maxbpg*

Indicates the maximum number of contiguous logical blocks any single file can allocate from a cylinder group before it is forced to begin allocating blocks from another cylinder group. Typically this value is set to approximately one quarter of the total contiguous logical blocks in a cylinder group. The intent is to prevent any single file from using up all the blocks in a single cylinder group, thus degrading access times for all files subsequently allocated in that cylinder group.

The effect of this limit is to cause big files to do long seeks more frequently than if they were allowed to allocate all the blocks in a cylinder group before seeking elsewhere. For file systems with exclusively large files, this parameter should be set higher.

`-m` *minfree*

Specifies the minimum free space threshold, or the percentage of space held back from normal users. This value can be set to `0`. However, up to a factor of three in throughput will be lost over the performance obtained at a 10% threshold. *Note:* If the value is raised above the current usage level, users will be unable to allocate files until enough files have been deleted to get under the higher threshold.

-o *space* | *time* The file system can either be instructed to try to minimize the *time* spent allocating blocks, or to try to minimize the *space* fragmentation on the disk. The default is *time*.

Generally, you should optimize for time unless the file system is over 90% full.

使用法

See `largefile(5)` for the description of the behavior of `tunefs` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`mkfs_ufs(1M)`, `newfs(1M)`, `attributes(5)`, `largefile(5)`

名前

txzonemgr – Trusted Extensions Zone Manager 構成ユーティリティー

形式

`/usr/sbin/txzonemgr [-c | -d[f] | -Z file]`

機能説明

`txzonemgr` シェルスクリプトは、Trusted Extensions が有効になっているシステムのラベル付きゾーンを作成、インストール、初期化、および起動するための、メニューベースの簡易 GUI ウィザードを提供します。また、このスクリプトはネットワークオプションやネームサービスオプションのメニュー項目、および大域ゾーンを既存の LDAP サーバーのクライアントにするためのメニュー項目も提供します。デフォルトではどのゾーンも、大域ゾーンと同じネームサービスと IP アドレスを使用するように構成されます。

`txzonemgr` は、後述する適切なオプションを使用することで、GUI モードで動作する代わりに、コマンド行モードで限られた数のコマンドを実行することもできます。

`txzonemgr` は、Zone Management 権利プロファイルで付与された役割、または大域ゾーン内のルートのいずれかによって実行されます。

オプション

オプションが指定されなかった場合、`txzonemgr` はメニューベースの GUI として動作します。オプションのいずれかを使用した場合、必ずコマンド行での動作になります。後述する場合を除き、複数のオプションを組み合わせることはできません。

次のオプションがサポートされています。

`-c`
: デフォルトのゾーンを作成します。ゾーンがまだ 1 つも存在していない必要があります。作成されるゾーンの数、名前、および属性は変更される可能性があります。ただし通常は、「public」ゾーンが作成されるほか、別のゾーンが少なくとも 1 つ作成されます。

`-d`
: すべてのゾーンを破棄します。すべてのゾーンが停止、アンインストール、および削除されます。`-f` オプションが一緒に指定されない場合に、確認プロンプトを表示します。

`-f`
: 強制実行。`-d` と組み合わせて使用すると、確認プロンプトを無効化できます。

`-z` *file*
: ファイルに指定された一覧を使用して、ゾーンセットを作成します。ファイル内の各行には、ゾーン名と対応するラベルを空白で区切って指定する必要があります。 ラベルに空白が含まれる場合でも、引用符で囲まないでください。 「スナップショット」という名前の既存のゾーンが存在する場合は、そのゾーンから指定されたすべてのゾーンのクローンが作成されます。それ以外の場合は、一覧の最初のゾーンがインストールされ、そのゾーンからスナップショットゾーンのクローンが作成されます。その後、スナップショットから残りのゾーンのクローンが作成されます。 ゾーンは手動ブートが必須となるように構成されてい

ます。 ただし、承認ユーザーが有効なラベルを GNOME 作業スペースに割り当てると、対応するゾーンが自動的にブートされます。

終了ステータス

GUI モードでは値は返されません。コマンド行動作のときに返される終了値を、次に示します。

`0`
正常終了。

`1`
エラーが発生しました。

`2`
無効な使用法。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
コマンドインタフェースの安定性	確実
対話型ダイアログ	インタフェースではない

関連項目

`zenity`(1) (not a SunOS man page), `ipadm(1M)`, `zoneadm(1M)`, `zonecfg(1M)`, `attributes(5)`, `rbac(5)`, `zones(5)`

Solaris Trusted Extensions 管理者の手順

注意事項

JDS からゾーンを管理する場合は、CDE アクションの代わりに `txzonemgr` を使用してください。

`txzonemgr` は `zenity` コマンドを使用します。詳細は、SunOS コレクションに含まれていない `zenity`(1) のマニュアルページを参照してください。

名前

tzreload – notify timezone update

形式

```
/usr/sbin/tzreload
```

機能説明

The `tzreload` command notifies active (running) processes to reread timezone information. The timezone information is cached in each process, absent a `tzreload` command, is never reread until a process is restarted. In response to a `tzreload` command, active processes reread the current timezone information at the next call to `ctime(3C)` and `mktime(3C)`. The `tzreload` notification is sent to processes within the current zone.

`tzreload` causes processes which are using the system timezone (in `/etc/default/init`) to reread the contents of that file.

In addition to notifying active processes, the `tzreload` command also notifies `cron(1M)`, to reinitialize the job scheduler with the new timezone information.

ファイル

`/usr/share/lib/zoneinfo`
: Standard zone information directory.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`cron(1M)`, `zdump(1M)`, `zic(1M)`, `ctime(3C)`, `mktime(3C)`, `attributes(5)`, `smf(5)`

注意事項

Although `tzreload` reinitializes `cron(1M)`, applications that are affected by timezone changes still need to be restarted or reinitialized if they do not reread the new timezone information before timezone changes take place.

The `timezone` update service is managed by the service management facility, `smf(5)`, under the service identifier:

```
svc:/system/timezone:default
```

Administrative actions on this service, such as enabling, disabling, or requesting refresh, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

Refreshing this service causes the `tzreload` command to be run, notifying running processes to reload their `timezone` caches. Disabling this service is not recommended.

名前 tzselect – select a time zone

形式 `/usr/bin/tzselect`

機能説明 The `tzselect` program asks you a series of questions about the current location and outputs the resulting time zone description to standard output. The output is suitable as a value for the `TZ` environment variable.

All user interaction is through standard input and standard error.

オプション The `tzselect` command has no options.

終了ステータス The following exit values are returned:

`0` Timezone information was successfully obtained.

`>0` An error occurred.

ファイル `/usr/share/lib/zoneinfo` directory containing timezone data files

属性 See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目 `zdump(1M)`, `zic(1M)`, `ctime(3C)`, `attributes(5)`

名前

uadmin – administrative control

形式

`/usr/sbin/uadmin` *cmd fcn* [*mdep*]

機能説明

The `uadmin` command provides control for basic administrative functions. This command is tightly coupled to the system administration procedures and is not intended for general use. It may be invoked only by the super-user.

Both the *cmd* (command) and *fcn* (function) arguments are converted to integers and passed to the `uadmin` system call. The optional *mdep* (machine dependent) argument is only available for the *cmd* values of `1` (`A_REBOOT`), `2` (`A_SHUTDOWN`), or `5` (`A_DUMP`). For any other *cmd* value, no *mdep* command-line argument is allowed.

When passing an *mdep* value that contains whitespaces, the string must be grouped together as a single argument enclosed within quotes, for example:

```
uadmin 1 1 “-s kernel/unix”
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`halt(1M)`, `reboot(1M)`, `uadmin(2)`, `attributes(5)`

警告

Shutting down or halting the system by means of `uadmin` does not update the boot archive. Avoid using this command after

- editing of files such as `/etc/system`
- installing new driver binaries or kernel binaries
- updating existing driver binaries or kernel binaries.

Use `reboot(1M)` or `halt(1M)` instead.

名前

ucodeadm – update processor microcode

形式

`/usr/sbin/ucodeadm -v`

`/usr/sbin/ucodeadm -u`*microcode-text-file*

`/usr/sbin/ucodeadm -i [-R` *path*`]` *microcode-text-file*

機能説明

The `ucodeadm` utility can be used to report running microcode revision on the processors, update microcode, or install microcode on the target system to be used during the boot process.

The *microcode-text-file* can be obtained from processor vendors.

オプション

`-v`	Report microcode revision.
`-u` *microcode-text-file*	Update microcode on all cross-call interrupt ready processors.
`-i` *microcode-text-file*	Install microcode files on target system to be used during the next boot cycle. The text file name must have the vendor name prefix, such as "`intel`" or "`amd`". By default the microcode files will be installed at: `/platform/i86pc/ucode/$VENDORSTR/` where `VENDORSTR` is either "`GenuineIntel`" or "`AuthenticAMD`".
`-R` *alternate path*	Install *microcode* path in the *alternate path*.

使用例

例 1 Reporting the Microcode Revision

The following example displays the microcode revision that is currently running:

```
# ucodeadm -v
```

例 2 Updating the Processor Microcode

The following example updates the processor microcode to `intel-ucode.txt`:

```
# ucodeadm -u intel-ucode.txt
```

例 3 Installing the Microcode on the Target System

The following example installs the microcode on the target system, `/export/ucode-path`:

```
# ucodeadm -i -R /export/ucode-path intel-ucode.txt
```

If an alternate path is used when installing the microcode on the target system, the installed microcode file is not used on the next boot cycle.

終了ステータス

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

psradm(1M), psrinfo(1M), attributes(5)

名前

ufsdump – incremental file system dump

形式

`/usr/sbin/ufsdump [options] [arguments] files_to_dump`

機能説明

`ufsdump` backs up all files specified by *files_to_dump* (usually either a whole file system or files within a file sytem changed after a certain date) to magnetic tape or disk file.

The `ufsdump` command can only be used on unmounted file systems, or those mounted read-only. Attempting to dump a mounted, read-write file system might result in a system disruption or the inability to restore files from the dump. Consider using the `fssnap(1M)` command to create a file system snapshot if you need a point-in-time image of a file system that is mounted.

If a filesystem was mounted with the logging option, it is strongly recommended that you run `ufsdump` as the root user. Running the command as a non-root user might result in the creation of an inconsistent dump.

options is a single string of one-letter `ufsdump` options.

arguments may be multiple strings whose association with the options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on.

files_to_dump is required and must be the last argument on the command line. See `OPERANDS` for more information.

With most devices `ufsdump` can automatically detect the end-of-media. Consequently, the `d`, `s`, and `t` options are not necessary for multi-volume dumps, unless `ufsdump` does not understand the way the device detects the end-of-media, or the files are to be restored on a system with an older version of the `restore` command.

オプション

The following options are supported:

`0–9`
: The “dump level.” All files specified by *files_to_dump* that have been modified since the last `ufsdump` at a lower dump level are copied to the *dump_file* destination (normally a magnetic tape device). For instance, if a “level `2`” dump was done on Monday, followed by a “level `4`” dump on Tuesday, a subsequent “level `3`” dump on Wednesday would contain all files modified or added since the “level `2`” (Monday) backup. A “level `0`” dump copies the entire file system to the *dump_file*.

`a` *archive_file*
: Archive file. Archive a dump table-of-contents in the specified *archive_file* to be used by `ufsrestore(1M)` to determine whether a file is in the dump file that is being restored.

`b` *factor*
: Blocking factor. Specify the blocking factor for tape writes. The default is 20 blocks per write for tapes of density less than 6250BPI (bytes-per-inch). The default blocking factor for tapes of density 6250BPI and greater is 64. The default blocking factor for cartridge

tapes (c option) is 126. The highest blocking factor available with most tape drives is 126. Note: the blocking factor is specified in terms of 512-byte blocks, for compatibility with tar(1).

c

Cartridge. Set the defaults for cartridge instead of the standard half-inch reel. This sets the density to 1000BPI and the blocking factor to 126. Since ufsdump can automatically detect the end-of-media, only the blocking parameter normally has an effect. When cartridge tapes are used, and this option is *not* specified, ufsdump will slightly miscompute the size of the tape. If the b, d, s or t options are specified with this option, their values will override the defaults set by this option.

d *bpi*

Tape density. Not normally required, as ufsdump can detect end-of-media. This parameter can be used to keep a running tab on the amount of tape used per reel. The default density is 6250BPI except when the c option is used for cartridge tape, in which case it is assumed to be 1000BPI per track. Typical values for tape devices are:

1/2 inch tape
6250 BPI

1/4 inch cartridge
1000 BPI The tape densities and other options are documented in the st(7D) man page.

D

Diskette. Obsolete option.

f *dump_file*

Dump file. Use *dump_file* as the file to dump to, instead of /dev/rmt/0. If *dump_file* is specified as –, dump to standard output.

If the name of the file is of the form *machine*:*device,* the dump is done from the specified machine over the network using rmt(1M). Since ufsdump is normally run by root, the name of the local machine must appear in the /.rhosts file of the remote machine. If the file is specified as *user*@*machine*:*device,* ufsdump will attempt to execute as the specified user on the remote machine. The specified user must have a .rhosts file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

l

Autoload. When the end-of-tape is reached before the dump is complete, take the drive offline and wait up to two minutes for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not, prompt for another tape and wait.

`L` *string*

Sets the tape label to *string*, instead of the default `none`. *string* may be no more than sixteen characters long. If it is longer, it is truncated and a warning printed; the dump will still be done. The tape label is specific to the `ufsdump` tape format, and bears no resemblance to IBM or ANSI-standard tape labels.

`n`

Notify all operators in the `sys` group that `ufsdump` requires attention by sending messages to their terminals, in a manner similar to that used by the `wall(1M)` command. Otherwise, such messages are sent only to the terminals (such as the console) on which the user running `ufsdump` is logged in.

`N` *device_name*

Use *device_name* when recording information in `/etc/dumpdates` (see the `u` option) and when comparing against information in `/etc/dumpdates` for incremental dumps. The *device_name* provided can contain no white space as defined in `scanf(3C)` and is case-sensitive.

`o`

Offline. Take the drive offline when the dump is complete or the end-of-media is reached and rewind the tape. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically. This prevents another process which rushes in to use the drive, from inadvertently overwriting the media.

`s` *size*

Specify the *size* of the volume being dumped to. Not normally required, as `ufsdump` can detect end-of-media. When the specified size is reached, `ufsdump` waits for you to change the volume. `ufsdump` interprets the specified size as the length in feet for tapes and cartridges. The values should be a little smaller than the actual physical size of the media (for example, 425 for a 450-foot cartridge). Typical values for tape devices depend on the `c` option for cartridge devices:

1/2 inch tape
: 2300 feet

60-Mbyte 1/4 inch cartridge
: 425 feet

150-Mbyte 1/4 inch cartridge
: 700 feet

`S`

Size estimate. Determine the amount of space that is needed to perform the dump without actually doing it, and display the estimated number of bytes it will take. This is useful with incremental dumps to determine how many volumes of media will be needed.

`t` *tracks*
: Specify the number of tracks for a cartridge tape. Not normally required, as `ufsdump` can detect end-of-media. The default is 9 tracks. The `t` option is not compatible with the `D` option. Values for Sun-supported tape devices are:

  60-Mbyte 1/4 inch cartridge
  : 9 tracks

  150-Mbyte 1/4 inch cartridge
  : 18 tracks

`T` *time_wait*`[hms]`
: Sets the amount of time to wait for an `autoload` command to complete. This option is ignored unless the `l` option has also been specified. The default time period to wait is two minutes. Specify time units with a trailing `h` (for hours), `m` (for minutes), or `s` (for seconds). The default unit is minutes.

`u`
: Update the dump record. Add an entry to the file `/etc/dumpdates,` for each file system successfully dumped that includes the file system name (or *device_name* as specified with the `N` option), date, and dump level.

`v`
: Verify. After each tape is written, verify the contents of the media against the source file system. If any discrepancies occur, prompt for new media, then repeat the dump/verification process. The file system *must* be unmounted. This option cannot be used to verify a dump to standard output.

`w`
: Warning. List the file systems that have not been backed up within a day. This information is gleaned from the files `/etc/dumpdates` and `/etc/vfstab`. When the `w` option is used, all other options are ignored. After reporting, `ufsdump` exits immediately.

`W`
: Warning with highlight. Similar to the `w` option, except that the `W` option includes all file systems that appear in `/etc/dumpdates`, along with information about their most recent dump dates and levels. File systems that have not been backed up within a day are highlighted.

オペランド

The following operand is supported:

files_to_dump
: Specifies the files to dump. Usually it identifies a whole file system by its raw device name (for example, `/dev/rdsk/c0t3d0s6`). Incremental dumps (levels `1` to `9`) of files changed after a certain date only apply to a whole file system. Alternatively, *files_to_dump* can identify individual files or directories. All named directories that may be examined by the user running `ufsdump`, as well as any explicitly-named files, are dumped. This dump is equivalent to a level `0` dump of the indicated portions of the filesystem, except that

`/etc/dumpdates` is not updated even if the `-u` option has been specified. In all cases, the files must be contained in the same file system, and the file system must be local to the system where `ufsdump` is being run.

files_to_dump is required and must be the last argument on the command line.

If no *options* are given, the default is `9uf /dev/rmt/0` *files_to_dump*.

使用法

See `largefile(5)` for the description of the behavior of `ufsdump` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

使用例

例 1 Using ufsdump

The following command makes a full dump of a root file system on `c0t3d0`, on a 150-MByte cartridge tape unit `0`:

```
example# ufsdump 0cfu /dev/rmt/0 /dev/rdsk/c0t3d0s0
```

The following command makes and verifies an incremental dump at level `5` of the `usr` partition of `c0t3d0`, on a 1/2 inch reel tape unit `1,`:

```
example# ufsdump 5fuv /dev/rmt/1 /dev/rdsk/c0t3d0s6
```

終了ステータス

While running, `ufsdump` emits many verbose messages. `ufsdump` returns the following exit values:

`0`
: Normal exit.

`1`
: Startup errors encountered.

`3`
: Abort – no checkpoint attempted.

ファイル

`/dev/rmt/0`
: default unit to dump to

`/etc/dumpdates`
: dump date record

`/etc/group`
: to find group `sys`

`/etc/hosts`
: to gain access to remote system with drive

`/etc/vfstab`
: list of file systems

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

cpio(1), tar(1), dd(1M), devnm(1M), fssnap(1M), prtvtoc(1M), rmt(1M), shutdown(1M), ufsrestore(1M), volcopy(1M), wall(1M), scanf(3C), ufsdump(4), attributes(5), largefile(5), st(7D)

注意事項

Read Errors

Fewer than 32 read errors on the file system are ignored.

Process Per Reel

Because each reel requires a new process, parent processes for reels that are already written hang around until the entire tape is written.

Operator Intervention

ufsdump requires operator intervention on these conditions: end of volume, end of dump, volume write error, volume open error or disk read error (if there are more than a threshold of 32). In addition to alerting all operators implied by the n option, ufsdump interacts with the operator on ufsdump's control terminal at times when ufsdump can no longer proceed, or if something is grossly wrong. All questions ufsdump poses *must* be answered by typing yes or no, as appropriate.

Since backing up a disk can involve a lot of time and effort, ufsdump checkpoints at the start of each volume. If writing that volume fails for some reason, ufsdump will, with operator permission, restart itself from the checkpoint after a defective volume has been replaced.

Suggested Dump Schedule

It is vital to perform full, “level 0”, dumps at regular intervals. When performing a full dump, bring the machine down to single-user mode using shutdown(1M). While preparing for a full dump, it is a good idea to clean the tape drive and heads. Incremental dumps should be performed with the system running in single-user mode.

Incremental dumps allow for convenient backup and recovery of active files on a more frequent basis, with a minimum of media and time. However, there are some tradeoffs. First, the interval between backups should be kept to a minimum (once a day at least). To guard against data loss as a result of a media failure (a rare, but possible occurrence), capture active files on (at least) two sets of dump volumes. Another consideration is the desire to keep unnecessary duplication of files to a minimum to save both operator time and media storage. A third consideration is the ease with which a particular backed-up version of a file can be located and restored. The following four-week schedule offers a reasonable tradeoff between these goals.

```
            Sun    Mon    Tue    Wed    Thu    Fri
Week 1:     Full    5      5      5      5      3
Week 2:            5      5      5      5      3
Week 3:            5      5      5      5      3
Week 4:            5      5      5      5      3
```

Although the Tuesday through Friday incrementals contain “extra copies” of files from Monday, this scheme assures that any file modified during the week can be recovered from the previous day's incremental dump.

Process Priority of ufsdump

`ufsdump` uses multiple processes to allow it to read from the disk and write to the media concurrently. Due to the way it synchronizes between these processes, any attempt to run dump with a `nice` (process priority) of ‘–5' or better will likely make `ufsdump` run *slower* instead of faster.

Overlapping Partitions

Most disks contain one or more overlapping slices because slice 2 covers the entire disk. The other slices are of various sizes and usually do not overlap. For example, a common configuration places `root` on slice 0, `swap` on slice 1, `/opt` on slice 5 and `/usr` on slice 6.

It should be emphasized that `ufsdump` dumps one `ufs` file system at a time. Given the above scenario where slice 0 and slice 2 have the same starting offset, executing `ufsdump` on slice 2 with the intent of dumping the entire disk would instead dump only the `root` file system on slice 0. To dump the entire disk, the user must dump the file systems on each slice separately.

使用上の留意点

The `/etc/vfstab` file does not allow the desired frequency of backup for file systems to be specified (as `/etc/fstab` did). Consequently, the `w` and `W` options assume file systems should be backed up daily, which limits the usefulness of these options.

名前

ufsrestore – incremental file system restore

形式

```
/usr/sbin/ufsrestore i | r | R | t | x [abcdfhlmostvyLT]
      [archive_file] [factor] [dumpfile] [n] [label]
      [timeout] [filename]...
```

機能説明

The `ufsrestore` utility restores files from backup media created with the `ufsdump` command. `ufsrestores`'s actions are controlled by the *key* argument. The *key* is exactly one `function` *letter* (`i`, `r`, `R` , `t`, or `x`) and zero or more *function modifier*s (letters). The *key* string contains no SPACE characters. Function modifier arguments are listed on the command line in the same order as their corresponding function modifiers appear in the *key* string.

filename arguments which appear on the command line, or as arguments to an interactive command, are treated as shell `glob` patterns by the `x` and `t` functions; any files or directories matching the patterns are selected. The metacharacters `*`, `?`, and `[ ]` must be protected from the shell if they appear on the command line. There is no way to quote these metacharacters to explicitly match them in a filename.

The temporary files `rstdir*` and `rstmode*` are placed in `/tmp` by default. If the environment variable `TMPDIR` is defined with a non-empty value, that location is used instead of `/tmp`.

オプション

Function Letters

You must specify one (and only one) of the function letters listed below. Note that `i`, `x`, and `r` are intended to restore files into an empty directory. The `R` function is intended for restoring into a populated directory.

`i` Interactive. After reading in the directory information from the media, `ufsrestore` invokes a shell-like interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted. Restoration has the same semantics as `x` (see below). See `Interactive Commands`, below, for a description of available commands.

`r` Recursive. Starting with an empty directory and a level 0 dump, the `r` function recreates the filesystem relative to the current working directory, exactly as it appeared when the dump was made. Information used to restore incremental dumps on top of the full dump (for example, `restoresymtable`) is also included. Several `ufsrestore` runs are typical, one for each higher level of dump (0, 1, ..., 9). Files that were deleted between the level 0 and a subsequent incremental dump will not exist after the final restore. To completely restore a file system, use the `r` function restore the level 0 dump, and again for each incremental dump. Although this function letter is intended for a complete restore onto a new file system (one just created with `newfs(1M)`), if the file system contains files not on the backup media, they are preserved.

`R` Resume restoring. If an `r`-mode `ufsrestore` was interrupted, this function prompts for the volume from which to resume restoring and continues the restoration from where it was left off. Otherwise identical to `r`.

t Table of contents. List each *filename* that appears on the media. If no *filename* argument is given, the root directory is listed. This results in a list of all files on the media, unless the `h` function modifier is in effect. The table of contents is taken from the media or from the specified archive file, when the `a` function modifier is used. The `a` function modifier is mutually exclusive with the `x` and `r` function letters.

x Extract the named files from the media. Files are restored to the same relative locations that they had in the original file system.

If the *filename* argument matches a directory whose contents were written onto the media, and the `h` modifier is not in effect, the directory is recursively extracted, relative to the current directory, which is expected to be empty. For each file, the owner, modification time, and mode are restored (if possible).

If you omit the *filename* argument or specify `.`, the root directory is extracted. This results in the entire tape being extracted, unless the `h` modifier is in effect. . With the `x` function, existing files are overwritten and `ufsrestore` displays the names of the overwritten files. Overwriting a currently-running executable can have unfortunate consequences.

Use the `x` option to restore partial file system dumps, as they are (by definition) not entire file systems.

Function Modifiers

a *archive_file* Read the table of contents from *archive_file* instead of the media. This function modifier can be used in combination with the `t`, `i`, or `x` function letters, making it possible to check whether files are on the media without having to mount the media. When used with the `x` and interactive (`i`) function letters, it prompts for the volume containing the file(s) before extracting them.

b *factor* Blocking factor. Specify the blocking factor for tape reads. For variable length SCSI tape devices, unless the data was written with the default blocking factor, a blocking factor at least as great as that used to write the tape must be used; otherwise, an error will be generated. Note that a tape block is 512 bytes. Refer to the man page for your specific tape driver for the maximum blocking factor.

c Convert the contents of the media in 4.1BSD format to the new `ufs` file system format.

d Debug. Turn on debugging output.

f *dump_file* Use *dump_file* instead of `/dev/rmt/0` as the file to restore from. Typically *dump_file* specifies a tape drive. If *dump_file* is specified as '–', `ufsrestore` reads from the standard input. This allows `ufsdump(1M)` and `ufsrestore` to be used in a pipeline to copy a file system:

```
example# ufsdump 0f - /dev/rdsk/c0t0d0s7 \
 | (cd /home;ufsrestore xf -)
```

If the name of the file is of the form *machine*:*device*, the restore is done from the specified machine over the network using `rmt(1M)`. Since `ufsrestore` is normally run by root, the name of the local machine must appear in the `/.rhosts` file of the remote machine. If the file is specified as *user*@*machine*:*device*, `ufsrestore` will attempt to execute as the specified user on the remote machine. The specified user must have a `.rhosts` file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

h — Extract or list the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the tape.

l — Autoload. When the end-of-tape is reached before the restore is complete, take the drive off-line and wait up to two minutes (the default, see the `T` function modifier) for the tape drive to be ready again. This gives autoloading (stackloader) tape drives a chance to load a new tape. If the drive is ready within two minutes, continue. If it is not, prompt for another tape and wait.

L *label* — The label that should appear in the header of the dump file. If the labels do not match, `ufsrestore` issues a diagnostic and exits. The tape label is specific to the `ufsdump` tape format, and bears no resemblance to IBM or ANSI-standard tape labels.

m — Extract by inode numbers rather than by filename to avoid regenerating complete pathnames. Regardless of where the files are located in the dump hierarchy, they are restored into the current directory and renamed with their inode number. This is useful if only a few files are being extracted.

o — Offline. Take the drive off-line when the restore is complete or the end-of-media is reached and rewind the tape. In the case of some autoloading 8mm drives, the tape is removed from the drive automatically.

s *n* — Skip to the *n*th file when there are multiple dump files on the same tape. For example, the command:

```
example# ufsrestore xfs /dev/rmt/0hn 5
```

would position you to the fifth file on the tape when reading volume 1 of the dump. If a dump extends over more than one volume, all volumes except the first are assumed to start at position 0, no matter what "`s` *n*" value is specified.

If "s *n*" is specified, the backup media must be at BOT (beginning of tape). Otherwise, the initial positioning to read the table of contents will fail, as it is performed by skipping the tape forward *n*-1 files rather than by using absolute positioning. This is because on some devices absolute positioning is very time consuming.

T timeout [hms] — Sets the amount of time to wait for an autoload command to complete. This function modifier is ignored unless the l function modifier has also been specified. The default timeout period is two minutes. The time units may be specified as a trailing h (hours), m (minutes), or s (seconds). The default unit is minutes.

v — Verbose. ufsrestore displays the name and inode number of each file it restores, preceded by its file type.

y — Do not ask whether to abort the restore in the event of tape errors. ufsrestore tries to skip over the bad tape block(s) and continue as best it can.

Interactive Commands

ufsrestore enters interactive mode when invoked with the i function letters. Interactive commands are reminiscent of the shell. For those commands that accept an argument, the default is the current directory. The interactive options are:

add [*filename*] — Add the named file or directory to the list of files to extract. If a directory is specified, add that directory and its files (recursively) to the extraction list (unless the h modifier is in effect).

cd *directory* — Change to *directory* (within the dump file).

delete [*filename*] — Delete the current directory, or the named file or directory from the list of files to extract. If a directory is specified, delete that directory and all its descendents from the extraction list (unless the h modifier is in effect). The most expedient way to extract a majority of files from a directory is to add that directory to the extraction list, and then delete specific files to omit.

extract — Extract all files on the extraction list from the dump media. ufsrestore asks which volume the user wishes to mount. The fastest way to extract a small number of files is to start with the last volume and work toward the first. If "s *n*" is given on the command line, volume 1 will automatically be positioned to file *n* when it is read.

help — Display a summary of the available commands.

ls [*directory*] — List files in *directory* or the current directory, represented by a '.' (period). Directories are appended with a '/' (slash). Entries marked for extraction are prefixed with a '*' (asterisk). If the verbose option is in effect, inode numbers are also listed.

marked [*directory*]	Like `ls`, except only files marked for extraction are listed.
pager	Toggle the pagination of the output from the `ls` and `marked` commands. The pager used is that defined by the `PAGER` environment variable, or more(1) if that envar is not defined. The `PAGER` envar may include white-space-separated arguments for the pagination program.
pwd	Print the full pathname of the current working directory.
quit	`ufsrestore` exits immediately, even if the extraction list is not empty.
setmodes	Prompts: `set owner/mode for '.'` (period). Type `y` for yes to set the mode (permissions, owner, times) of the current directory '.' (period) into which files are being restored equal to the mode of the root directory of the file system from which they were dumped. Normally, this is what you want when restoring a whole file system, or restoring individual files into the same locations from which they were dumped. Type `n` for no, to leave the mode of the current directory unchanged. Normally, this is what you want when restoring part of a dump to a directory other than the one from which the files were dumped.
setpager *command*	Sets the command to use for paginating output instead of the default or that inherited from the environment. The *command* string may include arguments in addition to the command itself.
verbose	Toggle the status of the `v` modifier. While `v` is in effect, the `ls` command lists the inode numbers of all entries, and `ufsrestore` displays information about each file as it is extracted.
what	Display the dump header on the media.

オペランド

The following operands are supported.

filename Specifies the pathname of files (or directories) to be restored to disk. Unless the `h` function modifier is also used, a directory name refers to the files it contains, and (recursively) its subdirectories and the files they contain. *filename* is associated with either the `x` or `t` function letters, and must come last.

使用法

See `largefile(5)` for the description of the behavior of `ufsrestore` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` An error occurred. Verbose messages are displayed.

環境変数

PAGER The command to use as a filter for paginating output. This can also be used to specify the options to be used. Default is `more(1)`.

TMPDIR　Selects the directory for temporary files. Defaults to `/tmp` if not defined in the environment.

ファイル

`/dev/rmt/0`	the default tape drive
`$TMPDIR/rstdir*`	file containing directories on the tape
`$TMPDIR/rstmode*`	owner, mode, and timestamps for directories
`./restoresymtable`	information passed between incremental restores

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`more(1)`, `mkfs(1M)`, `mount(1M)`, `rmt(1M)`, `ufsdump(1M)`, `ufsdump(4)`, `attributes(5)`, `largefile(5)`

診断

`ufsrestore` complains about bad option characters.

Read errors result in complaints. If `y` has been specified, or the user responds `y`, `ufsrestore` will attempt to continue.

If the dump extends over more than one tape, `ufsrestore` asks the user to change tapes. If the `x` or `i` function letter has been specified, `ufsrestore` also asks which volume the user wishes to mount. If the `s` modifier has been specified, and volume 1 is mounted, it is automatically positioned to the indicated file.

There are numerous consistency checks that can be listed by `ufsrestore`. Most checks are self-explanatory or can “never happen”. Common errors are given below.

`Converting to new file system format`
: A dump tape created from the old file system has been loaded. It is automatically converted to the new file system format.

filename`: not found on tape`
: The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, using a dump tape created on an active file system, or restoring a partial dump with the `r` function.

`expected next file` *inumber*`, got` *inumber*
: A file that was not listed in the directory showed up. This can occur when using a dump tape created on an active file system.

`Incremental tape too low`
: When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level has been loaded.

`Incremental tape too high`
: When doing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or one that has too high an incremental level has been loaded.

`media read error: invalid argument`
: Blocking factor specified for read is smaller than the blocking factor used to write data.

`Tape read error while restoring`
`Tape read error while skipping over inode inumber`
`Tape read error while trying to resynchronize`
`A tape read error has occurred`
: If a file name is specified, then its contents are probably partially wrong. If an inode is being skipped or the tape is trying to resynchronize, then no extracted files have been corrupted, though files may not be found on the tape.

`resync ufsrestore, skipped` *num*
: After a tape read error, `ufsrestore` may have to resynchronize itself. This message lists the number of blocks that were skipped over.

`Incorrect tape label. Expected 'foo', got 'bar'.`
: The `L` option was specified, and its value did not match what was recorded in the header of the dump file.

注意事項

`ufsrestore` can get confused when doing incremental restores from dump tapes that were made on active file systems.

A `level 0` dump must be done after a full restore. Because `ufsrestore` runs in user mode, it has no control over inode allocation. This means that `ufsrestore` repositions the files, although it does not change their contents. Thus, a full dump must be done to get a new set of directories reflecting the new file positions, so that later incremental dumps will be correct.

名前

unshare – ローカルシステムをリモートシステムからマウント不可能にする設定

形式

`unshare [-F protocol] [-a | [-t] [pathname | sharename]]`

機能説明

`unshare` コマンドは、ファイルシステムの共有を非公開にします。これにより、共有されているローカルのファイルシステムがファイル共有のための *protocol* で使用できなくなります。`-F` *protocol* オプションが省略された場合、`/etc/dfs/fstypes` ファイルで 1 番目にリストされているファイル共有プロトコルがデフォルトの共有プロトコルとして使用されます。

オプション

`-F` *protocol*
: ファイル共有プロトコルを識別します。

`-a`
: アクティブな共有をすべて非公開にします。

`-t`
: 一時的に共有を非公開にします。デフォルトでは、すべての共有は永続的に非公開にされています。

ファイル

`/etc/dfs/fstypes`
: ファイル共有プロトコルのリスト。デフォルトのファイル共有プロトコルは NFS です。

`/etc/dfs/sharetab`
: `share` コマンドによって公開されたローカルリソースのテーブルが含まれています。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	SUNWcs

関連項目

`share(1M)`, `shareall(1M)`, `attributes(5)`

注意事項

共有の情報で *pathname* または *resourcename* のいずれかが見つからない場合、標準エラー出力にエラーメッセージが送られます。

`unshare` コマンドが正常に終了すると、`unshare` コマンドに指定されたファイルシステムをマウントしているクライアントは、そのファイルシステムにアクセスできなくなります。

名前

unshare_nfs – make local NFS file systems unavailable for mounting by remote systems

形式

`unshare [-F nfs]` *pathname*

機能説明

The `unshare` command makes local file systems unavailable for mounting by remote systems. The shared file system must correspond to a line with NFS as the *FSType* in the file `/etc/dfs/sharetab`.

オプション

The following options are supported:

`-F` This option may be omitted if NFS is the first file system type listed in the file `/etc/dfs/fstypes`.

ファイル

`/etc/dfs/fstypes`

`/etc/dfs/sharetab`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

関連項目

`nfsd(1M)`, `share(1M)`, `attributes(5)`

注意事項

If the file system being unshared is a symbolic link to a valid pathname, the canonical path (the path which the symbolic link follows) will be unshared.

For example, if `/export/foo` is a symbolic link to `/export/bar` (`/export/foo -> /export/bar`), the following `unshare` command will result in `/export/bar` as the unshared pathname (and not `/export/foo`):

```
example#  unshare -F nfs /export/foo
```

For file systems that are accessed by NFS Version 4 clients, once the unshare is complete, all NFS Version 4 state (open files and file locks) are released and unrecoverable by the clients. If the intent is to share the file system after some administrative action, the NFS daemon (`nfsd`) should first be stopped and then the file system unshared. After the administrative action is complete, the file system would then be shared and the NFS daemon restarted. See `nfsd(1M)`

参 照

システム管理コマンド - パート 3

名前

update_drv – デバイスドライバ属性の変更

形式

```
update_drv [-f | -v] [-n] driver_module

update_drv [-b basedir] [-f | -v] [-n] -a [-m 'permission']
     [-i 'identify-name'] [-P 'privilege'] [-p 'policy'] driver_module

update_drv [-b basedir] [-f | -v] [-n] -d [-m 'permission']
     [-i 'identify-name'] [-P 'privilege'] [-p 'policy'] driver_module
```

機能説明

update_drv コマンドは、インストールされているデバイスドライバへの属性変更に関する情報をシステムに通知します。このコマンドを使用すると、driver.conf(4) ファイルを再読み込みしたり、ドライバマイナーノードのアクセス権や別名を追加、変更、または削除したりすることができます。

オプションを指定しないと、update_drv は driver.conf ファイルを再読み込みします。

別名が正常に更新された場合、ドライバのバインドは、再構成ブートまたはデバイスのホットプラグで有効になります。

アクセス権が正常に更新された場合は、新しいドライバマイナーノードのみが、変更された一連のファイルアクセス権を使用して作成されます。 既存のドライバマイナーノードは変更されません。

オプション

次のオプションがサポートされています。

-a
: アクセス権、別名、特権、またはポリシーのエントリを追加します。

  -a オプションを指定した場合は、アクセス権エントリ (-m オプションを使用)、ドライバの別名エントリ (-i オプションを使用)、デバイス特権 (-P オプションを使用)、またはデバイスポリシー (-p オプションを使用) を追加または更新できます。一致する (同じドライバ名とマイナーノードを持つ) マイナーノードアクセス権エントリが検出された場合は、置き換えられます。一致する (異なるドライバ名と同じ別名を持つ) 別名エントリが検出された場合は、エラーが報告されます。

  -a オプションと -d オプションは相互に排他的です。

-b *basedir*
: ドライバを、update_drv を実行しているシステムにインストールするのではなく、basedir というルートディレクトリを持つシステムでインストールまたは変更します。

  注 – -b オプションを使って、非大域ゾーンのルートファイルシステムを参照しないでください。この操作によって、大域ゾーンのファイルシステムを損傷したり、大域ゾーンのセキュリティを損ねたり、非大域ゾーンのファイルシステムを損傷する可能性があります。zones(5) を参照してください。

-d
アクセス権、別名、特権、またはポリシーのエントリを削除します。

-d オプションでは、-m *permission*、-i *identify-name*、-P *privilege*、または -p *policy* オプションを指定する必要があります。

-d オプションと -a オプションは相互に排他的です。

エントリが存在しない場合、update_drv はエラーを返します。

-f
ドライバモジュールを読み込み解除できない場合でも、システムで強制的に driver.conf ファイルを再読み込みするようにします。詳細は、「注意事項」の節を参照してください。

このオプションを指定しないで、ドライバの別名を削除した場合、update_drv は、次回のブート時にバインドファイルを更新します。ただし、このドライバの別名のバインドを参照しているデバイスが 1 つ以上残っている場合はエラーを返します。-f オプションを指定した場合、update_drv はそのようなデバイスが残っていてもエラーを返しません。

-i '*identify-name*'
ドライバの別名をスペースで区切ったリスト。-a または -d オプションが指定されていない場合、このオプションは無視されます。*identify-name* 文字列は必須です。すべての別名を削除する必要がある場合は、rem_drv(1M) をお勧めします。

-m '*permission*'
デバイスドライバのデバイスノードに対するファイルシステムアクセス権のスペースで区切られたリストを指定します。-a または -d オプションが指定されていない場合、このオプションは無視されます。permission 文字列は必須です。

-n
device_driver のロードや接続を試みず、そのドライバのシステム構成ファイルの変更だけを行います。

-p '*policy*'
-a オプションを指定した場合、policy は、完全なデバイスポリシーのスペースで区切られたリストです。 -d オプションの場合、ポリシーは、マイナーデバイス指定のスペースで区切られたリストです。マイナーデバイス指定は、/etc/security/device_policy 内のエントリに対して正確に照合されます。つまり、ワイルドカードの照合は実行されません。

-P '*privilege*'
-a オプションを指定した場合、privilege は、追加のドライバ特権のコンマで区切られたリストです。-d オプションの場合、privilege は 1 つの特権です。これらの特権は /etc/security/extra_privs ファイルに追加されるか、またはこのファイルから削除されます。

-v
冗長。

使用例

例1 既存のマイナーアクセス権エントリの追加または変更

次のコマンドは、clone ドライバの既存のマイナーアクセス権エントリを追加または変更します。

```
example# update_drv -a -m 'llc1 777 joe staff' clone
```

例2 すべてのマイナーアクセス権エントリの削除

次のコマンドは、USB プリンタドライバである usbprn ドライバのすべてのマイナーアクセス権エントリを削除します。

```
example# update_drv -d -m '* 0666 root sys' usbprn
```

例3 ドライバ別名エントリの追加

次のコマンドは、ugen ドライバのドライバ別名エントリを usb459,20 の識別文字列を使用して追加します。

```
example# update_drv -a -i '"usb459,20"' ugen
```

例4 ohci ドライバのための driver.conf ファイルの再読み込み

次のコマンドは、driver.conf(4) ファイルを再読み込みします。

```
 example# update_drv ohci
```

例5 tcp ソケットを開くために自己定義の特権を必要にする

次のコマンドは、tcp ソケットを開くために自己定義の特権を必要にします。

```
example#   update_drv -a -P net_tcp -p \
     'write_priv_set=net_tcp read_priv_set=net_tcp' tcp
```

例6 パス指向の別名の確立

次のコマンドは、パス指向の別名を確立して、特定のドライバ qlt が強制的に特定のデバイスパスに使用されるようにします。

```
example#   update_drv -a -i '"/pci@8,600000/SUNW,qlc@4"' qlt
```

終了ステータス

次の終了値が返されます。

0
正常終了。

>0
エラーが発生。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

add_drv(1M), modunload(1M), rem_drv(1M), driver.conf(4), attributes(5), privileges(5)

注意事項

-a または -d オプションが指定されている場合、update_drv は driver.conf ファイルを再読み込みしません。

driver.conf ファイルを強制的に更新すると、ドライババイナリモジュールを再読み込みすることなく、driver.conf ファイルが再読み込みされます。この場合は、切り離せないデバイスが古い driver.conf ファイルのドライバ大域プロパティーを参照するのに対して、残りのドライバインスタンスは新しい driver.conf ファイル内の大域プロパティーを参照します。

別名を追加することができます。これにより、すでに別のドライバによって管理されているデバイスのドライバのバインドが変更されます。-a オプションを使用して強制的に更新すると、新しいドライバにバインドしようとし、それができない場合はエラーを報告します。-m、-i、-P、または -p オプションのうちの複数を指定した場合は、強制フラグによって別名またはアクセス権の変更が試みられます。これはほかの操作が失敗した場合でも実行され、その逆も同じです。

名前

useradd – システムでの新しいユーザーログインの管理

形式

```
useradd [-A authorization [,authorization...]]
      [-b base_dir] [-c comment] [-d dir] [-e expire]
      [-f inactive] [-g group] [-G group [,group]...]
      [-K key=value] [-m [-k skel_dir]] [-p projname]
      [-P profile [,profile...]] [-R role [,role...]]
      [-s shell] [-S repository] [-u uid [-o]] login

useradd -D [-A authorization [,authorization...]]
      [-b base_dir] [-s shell [-k skel_dir]] [-e expire]
      [-f inactive] [-g group] [-K key=value] [-p projname]
      [-P profile [,profile...]] [-R role [,role...]]
```

機能説明

useradd は、files および ldap リポジトリの passwd、shadow、および user_attr データベースに新しいユーザーを追加します。-A および -P オプションは、それぞれ承認とプロファイルをユーザーに割り当てます。-R オプションはユーザーに役割を割り当てます。-p オプションはユーザーにプロジェクトを関連付けます。-K オプションは、ユーザーの user_attr エントリに *key=value* ペアを追加します。-K オプションを複数使用して、複数の *key=value* ペアを追加できます。

また、useradd では必要に応じて、ユーザーの補助グループメンバーシップを作成したり (-G オプション)、ユーザーのホームディレクトリを作成したり (-m オプション) できます。新しいログインは、passwd(1) コマンドが実行されるまで、ロックされたままになります。

useradd -D に -s、-k、-g、-b、-f、-e、-A、-P、-p、-R、または -K オプション (またはこれらのオプションを任意に組み合わせたもの) を指定すると、それぞれ対応するフィールドのデフォルト値を設定できます。下記の -D オプションを参照してください。そのあとで useradd コマンドを -D オプションなしで実行すると、これらの引数が使用されます。

このコマンドで作成されるシステムファイルエントリには、1 行あたり 2048 文字の制限があります。複数のオプションに長い引数を指定する場合は、この制限を超える可能性があります。

useradd で設定するユーザー名は、passwd(4) で説明されている形式でなければなりません。これらの要件が満たされていない場合は、警告メッセージが表示されます。ユーザー名の要件については、passwd(4) を参照してください。

新しいユーザーを作成するには、管理者に User Management プロファイルが付与されている必要があります。passwd、shadow、および user_attr の各種フィールドの設定に必要な承認については、passwd(4)、shadow(4)、および user_attr(4) を参照してください。グループおよびプロジェクトの割り当てに必要な承認については、group(4) および project(4) を参照してください。

オプション

次のオプションがサポートされています。

-A *authorization*
: auth_attr(4) で定義されている、1 つまたは複数のコンマ区切りの承認。承認の grant 権を持っているユーザーまたは役割だけが、その承認をアカウントに割り当てることができます。

-b *base_dir*
: 新しいログインホームディレクトリのベースディレクトリ (-d オプションを参照)。新しいユーザーアカウントを作成する場合、-m オプションまたは -d オプションを同時に指定するときを除き、*base_dir* はすでに存在している必要があります。

-c *comment*
: 任意のテキスト文字列を指定できます。一般にはログインの簡単な説明文ですが、現在はユーザーのフルネームを表すフィールドとして使用されています。この情報は、ユーザーの passwd エントリに保存されます。

-d *dir*
: 新しいユーザーの auto_home エントリを指定します。パス /home/*username* は /etc/passwd に入ります。ユーザーが続けて /home/*username* を参照すると、オートマウンタはここに指定されているディレクトリを /home/*username* にマウントします。

  -d オプションの引数は *server*:*dir* として指定できます。ここで、*server* はホームディレクトリがあるマシンのホスト名、*dir* はユーザーのホームディレクトリへのパスです。サーバーがリモートホストの場合、ユーザーのログイン時にシステムがホームディレクトリをマウントするために、リモートホストにホームディレクトリを作成する必要があります。-m オプションが指定されていて、サーバー名が指定されていない場合、ホームディレクトリはコマンドが実行されたホストに作成されます。

-D
: group、base_dir、skel_dir、shell、inactive、expire、proj、projname、および key=value ペアのデフォルト値を表示します。-g、-b、-f、-e、-A、-P、-p、-R、または -K とともに -D オプションを使用すると、指定したフィールドのデフォルト値を設定できます。デフォルト値は次のとおりです。

  group
  : other (GID は 1)

  base_dir
  : /export/home

  skel_dir
  : /etc/skel

shell
: `/usr/bin/bash`

inactive
: `0`

expire
: NULL 文字列

auths
: NULL 文字列

profiles
: NULL 文字列

proj
: `3`

projname
: `default`

key=value (`user_attr(4)` に定義されているペア)
: 存在しない

roles
: NULL 文字列

`-e` *expire*
: ログインの有効期限を指定します。この日付を過ぎると、どのユーザーもこのログインにアクセスできなくなります。expire オプション引数の日付は、テンプレートファイル `/etc/datemsk` に含まれている日付形式のいずれかを使用して入力します。`getdate(3C)` を参照してください。

  選択した日付形式にスペースが含まれている場合は、引用符で囲む必要があります。たとえば、`10/6/90` または `October 6, 1990` と入力できます。NULL 値 (" ") を指定すると、期限切れのステータスは無効になります。このオプションは一時的なログインを作成する場合に役立ちます。

`-f` *inactive*
: ログイン ID が最後に使用された日から無効と宣言されるまでの、最大日数を指定します。通常は正の整数です。値 `0` を指定すると、このステータスは無効になります。

`-g` *group*
: 既存グループのグループ ID (整数) または名前 (文字列) を指定します。`-D` オプションが指定されていない場合、新しいユーザーの一次グループメンバーシップを定義します。デフォルトでは、デフォルトグループになります。このデフォルト値は、`useradd -D -g` *group* を呼び出すことで再設定できます。0 から 99 の GID は、Solaris オペレーティングシステムによる割り当て用として予約されています。

-G *group*
既存グループのグループ ID (整数) または名前 (文字列) を指定します。新しいユーザーの補助グループメンバーシップを定義します。*group* が -g オプションと -G オプションで重複して指定されている場合、その項目は無視されます。指定できるグループの数は NGROUPS_MAX 以下です。0 から 99 の GID は、Solaris オペレーティングシステムによる割り当て用として予約されています。

-K *key=value*
ユーザーの属性に追加する *key=value* ペアを指定します。-K オプションを複数使用して、複数の *key=value* ペアを追加できます。暗黙に指定される特定のキーオプション (-A、-P、-R、-p) の代わりに、汎用の -K オプションと適切なキーを使用できます。使用可能な *key=value* ペアのリストについては、user_attr(4) を参照してください。type キーは、このオプションに有効なキーではありません。キーを重複して指定することはできません。

-k *skel_dir*
新しいユーザーのログインのホームディレクトリにコピーできるスケルトン情報 (.profile など) を格納しているディレクトリを指定します。このディレクトリはすでに存在している必要があります。システムには、この目的のために使用できるスケルトンディレクトリとして /etc/skel が用意されています。

-m
新しいユーザーのホームディレクトリが存在していない場合は、新たに作成します。ホームディレクトリがすでに存在している場合、そのディレクトリには *group* (ユーザーの一次グループ) による読み取り権、書き込み権、および実行権が設定されている必要があります。-d オプションに指定されたサーバー名がリモートホストの場合、システムはホームディレクトリの作成を試みません。

ディレクトリがまだ存在しない場合は、新しい ZFS データセットが作成されます。大域ゾーンにデータセットが rpool/export/home/*username* として作成されます。大域ゾーン以外の場合、データセットは *ROOT-dataset*/export/home/*username* として作成されます。ZFS データセットのマウントポイントは、デフォルトでは /export/home/*username* です。-d *path* が指定され、それがローカルマシン上のパスである場合、データセットは指定された場所にマウントされます。ZFS スナップショットを作成してそれをプロモートするためのアクセス権がユーザーに委任されます。新しく作成されるデータセットは、その親から暗号化設定を継承します。暗号化されている場合は、そのラップ鍵を変更するためのアクセス権がユーザーに委任されます。

-o
重複した (一意でない) UID を許可します。

-P *profile*
prof_attr(4) で定義されている実行プロファイルを 1 つ以上、コンマで区切って指定します。

-p *projname*
追加するユーザーに関連付けるプロジェクトの名前を指定します。project(4) で定義されている *projname* フィールドを参照してください。

-R *role*
user_attr(4) で定義されている実行プロファイルを 1 つ以上、コンマで区切って指定します。役割をほかの役割に割り当てることはできません。

-s *shell*
ユーザーのログインシェルとして使用するプログラムのフルパス名を指定します。指定しない場合は、以前に -D -s オプションで構成された値にデフォルトが設定されます。-D -s でデフォルトが設定されていない場合は、/usr/bin/bash が使用されます。 *shell* の値は有効な実行可能ファイルでなければなりません。

-S *repository*
有効なリポジトリは、files と ldap です。リポジトリは、更新するネームサービスを指定します。デフォルトのリポジトリは files です。リポジトリが files の場合、承認、プロファイル、および役割が、別のネームサービスリポジトリに存在していても、files リポジトリ内のユーザーに割り当てることができます。リポジトリが ldap の場合、割り当て可能な属性はすべて ldap リポジトリに存在している必要があります。

-u *uid*
新しいユーザーの UID を指定します。この UID は、<sys/param.h> ヘッダーファイルで定義された MAXUID より小さく、かつ負でない 10 進整数でなければなりません。UID のデフォルトは、現在割り当てられている最大の値の次に大きい使用可能な (一意の) 値です。たとえば、UID 100、105、および 200 が割り当て済みの場合、次のデフォルトの UID 番号は 201 になります。0 から 99 の UID は、Solaris オペレーティングシステムによる割り当て用として予約されています。

ファイル

/etc/datemsk

/etc/passwd

/etc/shadow

/etc/group

/etc/skel

/usr/include/limits.h

/etc/user_attr

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

passwd(1), profiles(1), roles(1), users(1B), groupadd(1M), groupdel(1M), groupmod(1M), grpck(1M), logins(1M), pwck(1M), userdel(1M), usermod(1M), getdate(3C), auth_attr(4), group(4), passwd(4), prof_attr(4), project(4), user_attr(4), attributes(5)

診断

エラーが発生した場合、useradd はエラーメッセージを表示し、ゼロ以外のステータスで終了します。

次のメッセージは、指定された login がすでに使用されていることを示します。

```
UX: useradd: ERROR: login is already in use. Choose another.
```

次のメッセージは、-u オプションで指定された *uid* が一意でないことを示します。

```
UX: useradd: ERROR: uid uid is already in use. Choose another.
```

次のメッセージは、-g オプションで指定された *group* がまだ作成されていないことを示します。

```
UX: useradd: ERROR: group group does not exist. Choose another.
```

次のメッセージは、-u オプションで指定された *uid* が予約済み UID の範囲 (0 から 99) に含まれていることを示します。

```
UX: useradd: WARNING: uid uid is reserved.
```

次のメッセージは、-u オプションで指定された *uid* が <sys/param.h> で定義された MAXUID を超えていることを示します。

```
UX: useradd: ERROR: uid uid is too big. Choose another.
```

次のメッセージは、/etc/passwd または /etc/shadow ファイルが存在しないことを示します。

```
UX: useradd: ERROR: Cannot update system files - login cannot be created.
```

次のメッセージは、コマンドを実行したユーザーに、この操作を実行するための十分な承認がないことを示します。

```
UX: roleadd: ERROR: Permission denied.
```

次のメッセージは、useradd コマンドで無効なディレクトリが指定されたことを示します。

```
UX: invalid_directory is not a valid directory. Choose another.
```

注意事項

useradd ユーティリティーは、デフォルトの有効範囲または指定された有効範囲内にある passwd、shadow、group、project、および user_attr データベースに定義を追加します。このユーティリティーは、ユーザー名 (または役割) およびユーザー ID が一意であることと、指定されたグループ名が存在することを、外部のネームサービスと照合して確認します。

名前

userdel – システムからのユーザーのログインの削除

形式

`userdel [-r] [-S repository] login`

機能説明

`userdel` ユーティリティーは、システムからユーザーアカウントを削除し、システムファイルやファイルシステムにアカウント関連の適切な変更を加えます。

ユーザーを削除するには、管理者に User Management プロファイルが付与されている必要があります。

オプション

次のオプションがサポートされています。

`- r` システムからユーザーのホームディレクトリを削除します。このディレクトリは存在する必要があります。このコマンドが正常に実行されたあと、ホームディレクトリの下にあるファイルやディレクトリにはアクセスできなくなります。ユーザーのホームディレクトリとして作成された ZFS データセットが削除されます。そのユーザー用に追加された `auto_home` エントリが削除されます。

`-S` *repository* 有効なリポジトリは、`files` と `ldap` です。リポジトリは、更新するネームサービスを指定します。デフォルトのリポジトリは `files` です。

オペランド

次のオペランドがサポートされています。

login 削除される既存のログイン名。

終了ステータス

次の終了値が返されます。

`0` 正常終了。

`2` 無効なコマンド構文。`userdel` コマンドの使用法に関するメッセージが表示されます。

`6` 削除されるアカウントが存在しません。

`8` 削除されるアカウントが使用されています。

`10` `/etc/group` または `/etc/user_attr` ファイルを更新できません。ただし、ログインは `/etc/passwd` ファイルから削除されます。

`12` ホームディレクトリを削除できないか、またはそれ以外の変更を行うことができません。

ファイル

`/etc/passwd` システムパスワードファイル

`/etc/shadow` ユーザーの暗号化パスワードおよび関連情報を含むシステムファイル

`/etc/group` グループ定義を含むシステムファイル

`/etc/user_attr` 追加のユーザー属性を含むシステムファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

auths(1), passwd(1), profiles(1), roles(1), users(1B), groupadd(1M), groupdel(1M), groupmod(1M), logins(1M), roleadd(1M), rolemod(1M), useradd(1M), usermod(1M), passwd(4), prof_attr(4), user_attr(4), attributes(5)

注意事項

userdel ユーティリティーは、files または ldap リポジトリの group、passwd、shadow、および user_attr データベースにあるアカウント定義を削除します。

名前

usermod – システムにあるユーザーのログイン情報の変更

形式

```
usermod [-u uid [-o]] [-g group] [-G [+|-]group [, group...]]
      [-d dir [-m]] [-s shell] [-c comment] [-l new_name]
      [-f inactive] [-e expire]
      [-A [+|-]authorization [,authorization]]
      [-P [+|-]profile [,profile]] [-R [+|-]role [,role]]
      [-K key[+|-]=value] [-S repository] login
```

機能説明

usermod ユーティリティーは、システムにあるユーザーのログイン定義を変更します。指定されたログインの定義を変更し、システムファイルやファイルシステムにログイン関連の適切な変更を加えます。

このコマンドで作成されるシステムファイルエントリには、1 行あたり 512 文字の制限があります。複数のオプションに長い引数を指定すると、この制限を超える可能性があります。

既存のユーザーのセキュリティー属性を変更するには、管理者に User Security プロファイルが付与されている必要があります。既存のユーザーのセキュリティー属性以外の属性を変更するには、User Management プロファイルが必要です。passwd、shadow、および user_attr の各種フィールドの設定に必要な承認については、passwd(4)、shadow(4)、および user_attr(4) を参照してください。グループの割り当てに必要な承認については、group(4) を参照してください。

オプション

次のオプションがサポートされています。

-A [+|-]*authorization*
: auth_attr(4) で定義されている、コンマで区切られた 1 つ以上の承認。アカウントに承認を割り当てることができるのは、authorization に対する grant 権限を持つユーザーまたは役割だけです。これにより、既存の承認設定はすべて置き換えられます。承認リストが指定されていない場合は、既存の設定が削除されます。

  接頭辞 + は承認を既存の承認に追加します。 接頭辞 - は承認を既存の承認から削除します。 接頭辞を指定しない場合、*authorization* で既存の承認を置換します。

-c *comment*
: コメント文字列を指定します。*comment* には任意のテキスト文字列を指定できます。一般にはログインの簡単な説明文ですが、現在はユーザーのフルネームを表すフィールドとして使用されています。この情報は、ユーザーの passwd エントリに保存されます。

-d *dir*
: ユーザーの新しいホームディレクトリを指定します。このディレクトリは、デフォルトでは *base_dir/login* になります。ここで、*base_dir* は新しいログインホームディレクトリのベースディレクトリであり、login は新しいログインです。この指定に従って、ユーザーの auto_home エントリが作成または変更されます。

このオプションの引数は、*server*:*dir* という形式で指定します。ここで、*server* はホームディレクトリがあるマシンのホスト名、*dir* はユーザーのホームディレクトリへのパスです。サーバーがリモートホストの場合、ユーザーのログイン時にシステムがホームディレクトリをマウントするために、リモートホストにホームディレクトリを作成する必要があります。-m オプションが指定されていて、サーバー名が指定されていない場合、ホームディレクトリはコマンドが実行されたホストに作成されます。

-e *expire*
ログインの有効期限を指定します。この日付を過ぎると、どのユーザーもこのログインにアクセスできなくなります。expire オプション引数は、テンプレートファイル /etc/datemsk に含まれているいずれかの日付形式を使用して入力される日付です。getdate(3C) を参照してください。

たとえば、10/6/90 または October 6, 1990 と入力できます。“ '' の値によって、期限切れの日付のステータスが無効になります。

-f *inactive*
ログイン ID が無効と宣言される前に、そのログイン ID の使用の間に経過することが許される最大日数を指定します。正常な値は正の整数です。0 の値によって、ステータスが無効になります。

-g *group*
既存のグループの整数 ID または文字列名を指定します。これにより、ユーザーの一次グループメンバーシップが再定義されます。

-G [+|-]*group*
既存グループのグループ ID (整数) または名前 (文字列) を指定します。新しいユーザーの補助グループメンバーシップを定義します。group が -g オプションと -G オプションで重複して指定されている場合、その項目は無視されます。指定できるグループの数は NGROUPS_MAX 以下です。0 から 99 の GID は、Solaris オペレーティングシステムによる割り当て用として予約されています。

接頭辞 + はグループを既存のグループに追加します。 接頭辞 - はグループを既存のグループから削除します。 接頭辞を指定しない場合、*group* で既存のグループを置換します。

-K *key*[+|-]=*value*
ユーザーの既存の *key*=*value* ペア属性を置き換えるか、またはこれらの属性に追加します。複数の *key*=*value* ペアを置き換えるか、または追加するために、複数の -K オプションを使用できます。ただし、キーを繰り返さないでください。特定の暗黙のキーオプション (-A、 -P、 -R、 -p) の代わりに、適切なキーを含む一般的な -K オプションを使用できます。使用可能な *key* のリストについては、user_attr(4) を参照してください。これらのキーの値は通常、キーに関連したマニュアルページまたはその他のソースに記載されています。たとえば、project キーの値に関するガイダンスについては project(4) を参照してくだ

さい。キー defaultpriv および limitpriv の値のリストを表示するには、コマンド ppriv(1) を -v および -l オプションとともに使用します。値を指定しないと、既存のキーが削除されます。

キーワード type には値 role または normal を指定できます。値 role を使用すると、アカウントが通常ユーザーから役割に変わります。値 normal を使用すると、アカウントは通常ユーザーのままです。

role アカウントとしては、役割 (-R または *roles=value*) は存在できません。

接頭辞 + は値を既存の値に追加します。 接頭辞 - は値を既存の値から削除します。 接頭辞を指定しない場合、*value* で既存の値を置換します。

接頭辞 +/- の操作は、キー auths、 profiles、 roles、 project、limitpriv、および defaultpriv にのみ適用可能です。

-l *new_logname*
ユーザーの新しいログイン名を指定します。ユーザー名の要件については、passwd(4) を参照してください。

-m
ユーザーのホームディレクトリを、-d オプションで指定された新しいディレクトリに移動します。ディレクトリがすでに存在する場合、そのディレクトリは、*group* による読み取り/書き込み/実行のアクセス権を持っている必要があります。ここで、*group* はユーザーの一次グループです。-d オプションに指定されたサーバー名がリモートホストの場合、システムはホームディレクトリの作成を試みません。

ディレクトリがまだ存在しない場合は、新しい ZFS データセットが作成されます。大域ゾーンにデータセットが rpool/export/home/*rolename* として作成されます。大域ゾーン以外の場合、データセットは *ROOT-dataset*/export/home/*rolename* として作成されます。ZFS データセットのマウントポイントは、デフォルトでは /export/home/*rolename* です。 -d *path* が指定され、それがローカルマシン上のパスである場合、データセットは指定された場所にマウントされます。ZFS スナップショットを作成してそれをプロモートするためのアクセス権が役割に委任されます。新しく作成されるデータセットは、その親から暗号化設定を継承します。暗号化されている場合は、そのラップ鍵を変更するためのアクセス権が役割に委任されます。

-o
このオプションは、指定された UID を複製できるようにします (一意ではない)。

-P [+|-]*profile*
prof_attr(4) で定義された、コンマで区切られた 1 つ以上の権利プロファイル。これにより、user_attr(4) に既存のプロファイル設定がある場合は置き換えられます。空のプロファイルリストを指定すると、既存の設定が削除されます。

接頭辞 + はプロファイルを既存のプロファイルに追加します。 接頭辞 - はプロファイルを既存のプロファイルから削除します。 接頭辞を指定しない場合、*profile* で既存のプロファイルを置換します。

-R [+|-]*role*
コンマで区切られた 1 つ以上の役割 (roleadd(1M) を参照)。これにより、既存の役割設定はすべて置き換えられます。役割リストが指定されていない場合は、既存の設定が削除されます。

接頭辞 + は役割を既存の役割に追加します。 接頭辞 - は役割を既存の役割から削除します。 接頭辞を指定しない場合、*role* で既存の役割を置換します。

-s *shell*
ログイン時のユーザーのシェルとして使用されるプログラムのフルパス名を指定します。*shell* の値は有効な実行可能ファイルでなければなりません。

-S *repository*
有効なリポジトリは、files と ldap です。リポジトリは、更新するネームサービスを指定します。デフォルトのリポジトリは files です。リポジトリが files の場合、承認、プロファイル、および役割が、別のネームサービスリポジトリに存在していても、files リポジトリ内のユーザーに割り当てることができます。リポジトリが ldap の場合、割り当て可能な属性はすべて ldap リポジトリに存在している必要があります。

-u *uid*
ユーザーの新しい UID を指定します。この値は、<param.h> で定義されている MAXUID より小さい、負でない 10 進数の整数である必要があります。ユーザーのホームディレクトリに関連付けられた UID は、このオプションでは変更されません。その UID が chown(1) を使用して手動で再割り当てされるまで、ユーザーは自分のホームディレクトリにアクセスできません。

オペランド

次のオペランドがサポートされています。

login
変更される既存のログイン名。

使用例

例 1 ユーザーへの特権の割り当て

次のコマンドは、高精度の時刻に影響を与える特権を、ユーザーの初期の継承可能な一連の特権に追加します。

```
# usermod -K defaultpriv=basic,proc_clock_highres jdoe
```

このコマンドにより、user_attr に次のエントリが生成されます。

```
jdoe::::type=normal;defaultpriv=basic,proc_clock_highres
```

例 2 ユーザーの制限セットからの特権の削除

次のコマンドは、指定されたユーザーがディレクトリへのハードリンクを作成したり、ディレクトリのリンクを解除したりできるようにする特権を削除します。

```
# usermod -K limitpriv=all,!sys_linkdir jdoe
```

このコマンドにより、user_attr に次のエントリが生成されます。

```
jdoe::::type=normal;defaultpriv=basic,limitpriv=all,!sys_linkdir
```

例 3 ユーザーの基本セットからの特権の削除

次のコマンドは、指定されたユーザーがそのユーザーのセッションの外部にあるプロセスを検査できるようにする特権を削除します。

```
# usermod -K defaultpriv=basic,!proc_session jdoe
```

このコマンドにより、user_attr に次のエントリが生成されます。

```
jdoe::::type=normal;defaultpriv=basic,!proc_session;limitpriv=all
```

例 4 ユーザーへの役割の割り当て

次のコマンドは、ユーザーに役割を割り当てます。このコマンドを実行する前に、roleadd(1M) を使って役割を作成しておく必要があります。

```
# usermod -R mailadm jdoe
```

このコマンドにより、user_attr に次のエントリが生成されます。

```
jdoe::::type=normal;roles=mailadm;defaultpriv=basic;limitpriv=all
```

例 5 ユーザーからのすべてのプロファイルの削除

次のコマンドは、ユーザーに直接付与されたすべてのプロファイルを削除します。ユーザーは、policy.conf(4) の PROFS_GRANTED キーによって付与される権利プロファイルは引き続き保持します。

```
# usermod -P "" jdoe
```

終了ステータス

エラーが発生した場合、usermod はエラーメッセージを出力し、次のいずれかの値で終了します。

2
: コマンド構文が無効でした。usermod コマンドの使用法に関するメッセージが表示されます。

3
: オプションに無効な引数が指定されました。

4
: -u オプションで指定された *uid* はすでに使用されています。

5
パスワードファイルにエラーが含まれています。pwconv(1M) を使用すると、可能性があるエラーを修正できます。passwd(4) を参照してください。

6
変更されるログインが存在しないか、*group* が存在しないか、またはログインシェルが存在しません。

8
変更されるログインが使用されています。

9
この *new_logname* はすでに使用されています。

10
/etc/group または /etc/user_attr ファイルを更新できません。ほかの更新要求は実装されます。

11
ホームディレクトリを移動するための容量が不足しています (-m オプション)。ほかの更新要求は実装されます。

12
ホームディレクトリの新しいホームディレクトリへの移動を完了できません。

ファイル

/etc/group
グループ定義を含むシステムファイル

/etc/datemsk
日付形式のシステムファイル

/etc/passwd
システムパスワードファイル

/etc/shadow
ユーザーの暗号化パスワードおよび関連情報を含むシステムファイル

/etc/user_attr
追加のユーザーおよび役割属性を含むシステムファイル

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	確実

関連項目

chown(1), passwd(1), users(1B), groupadd(1M), groupdel(1M), groupmod(1M), logins(1M), pwconv(1M), roleadd(1M), roledel(1M), rolemod(1M), useradd(1M), userdel(1M), getdate(3C), auth_attr(4), passwd(4), policy.conf(4), prof_attr(4), user_attr(4), attributes(5)

注意事項

usermod ユーティリティーは、デフォルトの有効範囲または指定された有効範囲内にある passwd、shadow、group、project、および user_attr データベースにある定義を変更します。このユーティリティーは、ユーザー名およびユーザー ID が一意であることを、外部のネームサービスと照合して確認します。

usermod ユーティリティーは、日付の書式設定のために、SUNWaccr で使用可能な /etc/datemsk ファイルを使用します。

名前

utmpd – utmpx monitoring daemon

形式

```
utmpd [-debug]
```

機能説明

The utmpd daemon monitors the /var/adm/utmpx file. See utmpx(4) (and utmp(4) for historical information).

utmpd receives requests from pututxline(3C) by way of a named pipe. It maintains a table of processes and uses poll(2) on /proc files to detect process termination. When utmpd detects that a process has terminated, it checks that the process has removed its utmpx entry from /var/adm/utmpx. If the process' utmpx entry has not been removed, utmpd removes the entry. By periodically scanning the /var/adm/utmpx file, utmpd also monitors processes that are not in its table.

オプション

-debug
: Run in debug mode, leaving the process connected to the controlling terminal. Write debugging information to standard output.

終了ステータス

The following exit values are returned:

0
: Successful completion.

>0
: An error occurred.

ファイル

/etc/default/utmpd
: You can set default values for the flags listed below. For example: SCAN_PERIOD=600

  SCAN_PERIOD
  : The number of seconds that utmpd sleeps between checks of /proc to see if monitored processes are still alive. The default is 300.

  MAX_FDS
  : The maximum number of processes that utmpd attempts to monitor. The default value is 4096.

  WTMPX_UPDATE_FREQ
  : The number of seconds that utmpd sleeps between read accesses of the wtmpx file. The wtmpx file's last access time is used by init(1M) on reboot to determine when the operating system became unavailable. The default is 60.

/var/adm/utmpx
: File containing user and accounting information for commands such as who(1), write(1), and login(1).

/proc
: Directory containing files for processes whose utmpx entries are being monitored.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

svcs(1), init(1M), svcadm(1M), poll(2), pututxline(3C), proc(4), utmp(4), utmpx(4), attributes(5), smf(5)

注意事項

If the filesystem holding /var/adm/wtmpx is mounted with options which inhibit or defer access time updates, an unknown amount of error will be introduced into the utmp DOWN_TIME record's timestamp in the event of an uncontrolled shutdown (for example, a crash or loss of power). Controlled shutdowns will update the modify time of /var/adm/wtmpx, which will be used on the next boot to determine when the previous shutdown ocurred, regardless of access time deferral or inhibition.

The utmpd service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/utmp:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

uucheck – check the uucp directories and permissions file

形式

`/usr/lib/uucp/uucheck [-v] [-x debug-level]`

機能説明

`uucheck` checks for the presence of the `uucp` system required files and directories. `uucheck` also does error checking of the `Permissions` file (`/etc/uucp/Permissions`).

`uucheck` is executed during package installation. `uucheck` can only be used by the super-user or `uucp`.

オプション

The following options are supported:

`-v` Give a detailed (verbose) explanation of how the `uucp` programs will interpret the `Permissions` file.

`-x` *debug-level* Produce debugging output on the standard output. *debug-level* is a number from `0` to `9`. Higher numbers give more detailed debugging information.

ファイル

`/etc/uucp/Devices`

`/etc/uucp/Limits`

`/etc/uucp/Permissions`

`/etc/uucp/Systems`

`/var/spool/locks/*`

`/var/spool/uucp/*`

`/var/spool/uucppublic/*`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

`uucp(1C)`, `uustat(1C)`, `uux(1C)`, `uucico(1M)`, `uusched(1M)`, `attributes(5)`

使用上の留意点

The program does not check file/directory modes or some errors in the `Permissions` file such as duplicate login or machine name.

名前

uucico – file transport program for the uucp system

形式

```
/usr/lib/uucp/uucico [-f] [-c type] [-d spool-directory]
     [-i interface] [-r role-number] [-s system-name]
     [-x debug-level]
```

機能説明

uucico is the file transport program for uucp work file transfers.

オプション

The following options are supported:

-f	This option is used to "force execution" of uucico by ignoring the limit on the maximum number of uucicos defined in the /etc/uucp/Limits file.
-c *type*	The first field in the Devices file is the "Type" field. The -c option forces uucico to only use entries in the "Type" field that match the user specified type. The specified type is usually the name of a local area network.
-d *spool-directory*	This option specifies the directory *spool-directory* that contains the uucp work files to be transferred. The default spool directory is /var/spool/uucp.
-i *interface*	This option defines the *interface* used with uucico. The interface only affects slave mode. Known interfaces are UNIX (default), TLI (basic Transport Layer Interface), and TLIS (Transport Layer Interface with Streams modules, read/write).
-r *role-number*	The *role-number* 1 is used for master mode. *role-number* 0 is used for slave mode (default). When uucico is started by a program or cron, *role-number* 1 should be used for master mode.
-s *system-name*	The -s option defines the remote system (*system-name*) that uucico will try to contact. It is required when the role is master; *system-name* must be defined in the Systems file.
-x *debug-level*	Both uux and uucp queue jobs that will be transferred by uucico. These jobs are normally started by the uusched scheduler, for debugging purposes, and can be started manually. For example, the shell Uutry starts uucico with debugging turned on. The *debug-level* is a number between 0 and 9. Higher numbers give more detailed debugging information.

ファイル

/etc/uucp/Devconfig

/etc/uucp/Devices

/etc/uucp/Limits

/etc/uucp/Permissions

/etc/uucp/Sysfiles

/etc/uucp/Systems

/var/spool/locks/*

/var/spool/uucp/*

/var/spool/uucppublic/*

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

uucp(1C), uustat(1C), uux(1C), Uutry(1M), cron(1M), uusched(1M), attributes(5)

名前

uucleanup – uucp spool directory clean-up

形式

```
/usr/lib/uucp/uucleanup [-Ctime] [-Dtime] [-mstring]
     [-otime] [-ssystem] [-Wtime] [-xdebug-level] [-Xtime]
```

機能説明

uucleanup will scan the spool directories for old files and take appropriate action to remove them in a useful way:

- Inform the requester of send/receive requests for systems that can not be reached.
- Return undeliverable mail to the sender.
- Deliver rnews files addressed to the local system.
- Remove all other files.

In addition, there is a provision to warn users of requests that have been waiting for a given number of days (default 1 day). Note: uucleanup will process as if all option times were specified to the default values unless time is specifically set.

This program is typically started by the shell uudemon.cleanup, which should be started by cron(1M).

オプション

-Ctime	Remove any C. files greater or equal to time days old and send appropriate information to the requester (default 7 days).
-Dtime	Remove any D. files greater or equal to time days old, make an attempt to deliver mail messages, and execute rnews when appropriate (default 7 days).
-m*string*	Include *string* in the warning message generated by the -W option. The default line is "See your local administrator to locate the problem".
-otime	Delete other files whose age is more than time days (default 2 days).
-s*system*	Execute for *system* spool directory only.
-Wtime	Any C. files equal to time days old will cause a mail message to be sent to the requester warning about the delay in contacting the remote. The message includes the *JOBID*, and in the case of mail, the mail message. The administrator may include a message line telling whom to call to check the problem (-m option) (default 1 day).
-x*debug-level*	Produce debugging output on standard ouput. *debug-level* is a single digit between 0 and 9; higher numbers give more detailed debugging information. (This option may not be available on all systems.)
-Xtime	Any X. files greater or equal to time days old will be removed. The D. files are probably not present (if they were, the X. could get executed). But if there are D. files, they will be taken care of by D. processing (default 2 days).

ファイル

/usr/lib/uucp	directory with commands used by uucleanup internally
/var/spool/uucp	spool directory

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

uucp(1C), uux(1C), cron(1M), attributes(5)

名前

uusched – uucp file transport program scheduler

形式

`/usr/lib/uucp/uusched [-u debug-level] [-x debug-level]`

機能説明

`uusched` is the `uucp(1C)` file transport scheduler. It is usually started by the daemon *uudemon.hour* that is started by `cron(1M)` from an entry in user `uucp`'s crontab file:

```
11,41 * * * * /etc/uucp/uucp/uudemon.hour
```

オプション

The options are for debugging purposes only. *debug-level* is a number between `0` and `9`. Higher numbers give more detailed debugging information:

The following options are supported:

`-u` *debug-level* Passes the `-u` *debug-level* option `uucico(1M)` as `-x` *debug-level*.

`-x` *debug-level* Outputs debugging messages from `uusched`.

ファイル

`/etc/uucp/Devices`

`/etc/uucp/Permissions`

`/etc/uucp/Systems`

`/var/spool/locks/*`

`/var/spool/uucp/*`

`/var/spool/uucppublic/*`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

`uucp(1C)`, `uustat(1C)`, `uux(1C)`, `cron(1M)`, `uucico(1M)`, `attributes(5)`

名前

Uutry, uutry – attempt to contact remote system with debugging on

形式

`/usr/lib/uucp/Uutry [-r] [-c` *type*`] [-x` *debug-level*`]` *system-name*

機能説明

`Uutry` is a shell script that is used to invoke `uucico(1M)` to call a remote site. Debugging is initially turned on and is set to the default value of `5`. The debugging output is put in file `/tmp/`*system-name*.

オプション

The following options are supported:

`-r` This option overrides the retry time that is set in file `/var/uucp/.Status/`*system-name*.

`-c` *type* The first field in the `Devices` file is the "Type" field. The `-c` option forces `uucico` to use only entries in the "Type" field that match the user-specified `type`. The specified `type` is usually the name of a local area network.

`-x` *debug-level* *debug-level* is a number from `0` to `9`. Higher numbers give more detailed debugging information.

ファイル

`/etc/uucp/Devices`

`/etc/uucp/Limits`

`/etc/uucp/Permissions`

`/etc/uucp/Systems`

`/tmp/`*system-name*

`/var/spool/locks/*`

`/var/spool/uucp/*`

`/var/spool/uucppublic/*`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

`uucp(1C)`, `uux(1C)`, `uucico(1M)`, `attributes(5)`

名前

uuxqt – execute remote command requests

形式

`/usr/lib/uucp/uuxqt [-s system] [-x debug-level]`

機能説明

`uuxqt` is the program that executes remote job requests from remote systems generated by the use of the `uux` command. (`mail` uses `uux` for remote mail requests). `uuxqt` searches the spool directories looking for execution requests. For each request, `uuxqt` checks to see if all the required data files are available, accessible, and the requested commands are permitted for the requesting system. The `Permissions` file is used to validate file accessibility and command execution permission.

There are two environment variables that are set before the `uuxqt` command is executed:

- `UU_MACHINE` is the machine that sent the job (the previous one).
- `UU_USER` is the user that sent the job.

These can be used in writing commands that remote systems can execute to provide information, auditing, or restrictions.

オプション

The following options are supported:

`-s` *system*	Specifies the remote *system* name.
`-x` *debug-level*	*debug-level* is a number from `0` to `9`. Higher numbers give more detailed debugging information.

ファイル

`/etc/uucp/Limits`

`/etc/uucp/Permissions`

`/var/spool/locks/*`

`/var/spool/uucp/*`

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

関連項目

`mail(1)`, `uucp(1C)`, `uustat(1C)`, `uux(1C)`, `uucico(1M)`, `attributes(5)`

名前

vbiosd – vbios daemon

形式

```
vbiosd
```

機能説明

The vbiosd daemon is started at boot time to provide an execution environment for adapter-supplied VBIOS code that is used in graphics mode switching.

This mode switching occurs when exiting X, switching virtual terminals (see vtdaemon(1M) for more information), and during fast reboot.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Uncommitted

関連項目

svcs(1), svcadm(1M), vtdaemon(1M), attributes(5), smf(5)

注意事項

The vbios daemon is managed by the service management facility, smf(5), under the service identifier:

```
svc:/system/vbiosd:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

vdiskadm – 仮想ディスクの作成と管理

形式

```
vdiskadm create -s size [-t type[:opt],[opt]
     [-c comment] vdname

vdiskadm destroy [-r] vdname|snapshot

vdiskadm snapshot vdname@snapname

vdiskadm rollback [-r] snapshot

vdiskadm clone [-c comment] vdname|snapshot clone_vdname

vdiskadm move vdname dir

vdiskadm rename vdname|snapshot vdname|snapshot

vdiskadm list [-fp]vdname

vdiskadm verify vdname

vdiskadm prop-get [-l] -p property vdname

vdiskadm prop-set -p property=value vdname

vdiskadm prop-add -p property=value vdname

vdiskadm prop-del -p property vdname

vdiskadm import [-fnpqm] [-x type] -d file|zvol|dsk
     [-t type[:opt]] vdname

vdiskadm export -x type[:opt] -d file|zvol|dsk vdname

vdiskadm convert [-t type[:opt]] vdname

vdiskadm translate [-i type[:opt]] -I input_file -x type[:opt]
     -d output_file

vdiskadm help [command]
```

機能説明

`vdiskadm` コマンドは、`dom0` 内の仮想ディスクを管理します。前述の「形式」に含まれている *vdname* は仮想ディスクのパス名であり、その最大長は `MAXPATHLEN` (1024 バイト) です。

`vdiskadm` は一連のサブコマンドとして実装されており、その多くは独自のオプションやオペランドを備えています。これらのサブコマンドは、後述の「サブコマンド」で説明されています。

次の各項目では、仮想ディスクに関係する概念について説明します。

スナップショット

スナップショットとは、仮想ディスクの読み取り専用コピーのことです。スナップショットはきわめてすばやく作成することができ、最初は領域をほとんど消費しません。アクティブな仮想ディスク内のデータが変更されると、スナップショット用のデータが増加します。これは、本来ならアクティブな仮想ディスクと共有されていたデータです。

クローン

クローンとは、仮想ディスクの書き込み可能なコピーのことです。クローンのデフォルトタイプは、元の仮想ディスクのマージされた (つまり合体した) コピーになります。マージクローンの例としては、仮想ディスクがいくつかのスナップショットから構成されている場合が挙げられます。その後にクローン処理を行うと、新しい仮想ディスクにはスナップショットは一切含まれません。クローンのタイプは元の仮想ディスクと同じ (つまり `vmdk:fixed`) になります。マージクローンの作成時には、元の仮想ディスクやそのスナップショットに対するリンクは一切作成されません。このようにマージクローンにはリンクが存在しないため、別の物理マシンに移動できます。

数値

数値プロパティーの値は、人間が読み取ることのできるサフィックス (`k`、`KB`、`M`、`Gb` など) を使って指定できます。使用可能な最大サフィックスは `Z` (ゼタバイト) です。次の指定はすべて有効 (および等価) です。

```
1536M  1.5g  1.50GB
```

仮想ディスクのタイプ

サポートされている仮想ディスクのタイプとオプションは、次のとおりです。

- `vmdk:fixed`
- `vmdk:sparse`
- `vdi:fixed`
- `vdi:sparse`
- `vhd:fixed`
- `vhd:sparse`
- `raw:fixed`

ここで、`vmdk` はネイティブの VMware 形式、`vdi` はネイティブの VirtualBox 形式、`vhd` はネイティブの Microsoft 形式であり、`raw` は raw ディスクと思われるファイルを記述します。`raw` ディスクは常に fixed 形式ですが、これは、オプションの明示的な設定や暗黙的な認識を可能にするためです。

タイプが指定されなかった場合のデフォルト値は、`vmdk` です。オプションが指定されなかった場合のデフォルト値は、タイプ `raw` では `fixed`、タイプ `vmdk`、`vdi`、および `vhd` では `sparse` です。

ネイティブプロパティーとユーザー定義プロパティー

プロパティーは、ネイティブとユーザー定義の 2 つの種類に分けられます。ネイティブプロパティーは、内部統計データのエクスポートや `vdiskadm` の動作制御を行います。また、ネイティブプロパティーは編集可能であるか、読み取り専用です。ユーザー定義プロパティーは、`vdiskadm` の動作に何の影響も与えない任意の文字列です。これを使用すると、ユーザー環境で意味のある方法で仮想ディスクに注釈を付けることができます。ユーザー定義プロパティー名には、ネイティブプロパティーと区別するためにコロン (`:`) 文字を含める必要があります。

プロパティーは仮想ディスクにのみ関連付けられ、個々のスナップショットには関連付けられません。

すべての仮想ディスクには、さまざまな動作を制御するネイティブプロパティーに加え、仮想ディスクに関する統計データを出力するネイティブプロパティーがあります。

仮想ディスクのネイティブプロパティーを次に示します。

`cdrom`
: 仮想ディスクが CDROM の場合に真になるブール (論理) 型プロパティー。

`removable`
: 仮想ディスクがリムーバブルメディアの場合に真になるブール型プロパティー。

`readonly`
: 仮想ディスクが読み取り専用の場合に真になるブール型プロパティー。このプロパティーは読み取り専用です。

`sectors`
: 指定された仮想ディスク内のディスクセクター数を含む数値プロパティー。このプロパティーは読み取り専用です。

`name`
: 仮想ディスクの名前を表す文字列プロパティー。このプロパティーは読み取り専用です。

`max-size`
: 仮想ディスクの最大サイズ (バイト) を含む数値プロパティー。このプロパティーは読み取り専用です。

`effective-size`
: 仮想ディスクの実効サイズ (バイト) を含む数値プロパティー。実効サイズにはデータファイルとすべてのスナップショットのサイズが含まれます。実効サイズは最大サイズを超える可能性があります。このプロパティーは読み取り専用です。

`creation-time`
: 仮想ディスクが作成された日付と時刻を含む文字列プロパティー。このプロパティーは読み取り専用です。

`creation-time-epoch`
: creation-time プロパティーをエポックからの秒数 (1970 年 1 月 1 日 00:00:00 UTC からの秒数) として記述した数値プロパティー。このプロパティーは読み取り専用です。

`modification-time`
: 仮想ディスクが最後に変更された日付と時刻を含む文字列プロパティー。このプロパティーは読み取り専用です。

`modification-time-epoch`
: modification-time プロパティーをエポックからの秒数 (1970 年 1 月 1 日 00:00:00 UTC からの秒数) として記述した数値プロパティー。このプロパティーは読み取り専用です。

`description`
: 仮想ディスクの作成時またはクローン時に指定されたコメントを含む文字列プロパティー。このプロパティーは読み取り専用です。

`type`
: 仮想ディスクのタイプ (`vmdk`、`vhd`、`vdi`、`raw` のいずれか) を含む文字列プロパティー。このプロパティーは読み取り専用です。

`sparse`
: 仮想ディスクが `sparse` 形式の場合に真になるブール型プロパティー。このプロパティーは読み取り専用です。

`owner`
: 仮想ディスクの所有者のユーザー名を含む文字列プロパティー。このプロパティーは編集可能です。

サブコマンド

次の各項目では、`vdiskadm` サブコマンドとその引数について説明します。

vdiskadm create

```
vdiskadm create -s size [-t type[:opt],[opt]]
    [-c comment] vdname
```

指定されたサイズの新しい仮想ディスクを、*vdname* で指定された場所に作成します。仮想ディスクへのパスが *vdname* に含まれている場合、そのパスに付随するディレクトリが、仮想ディスクの作成時に作成されます。このサブコマンドには、次に示すオプションがあります。

`-t` *type*[:*opt*],[*opt*]]
: 作成される仮想ディスクのタイプを指定します。デフォルトのタイプは `vmdk` です。`vmdk` および `vdi` タイプの場合、デフォルトのオプションは `sparse` になります。タイプ `raw` の場合、デフォルトのオプションは `fixed` になります。

`-c` *comment*
: 仮想ディスクに添付可能なコメント。

vdiskadm destroy

```
vdiskadm destroy [-r] vdname|snapshot
```

指定された仮想ディスクまたはスナップショットを破棄します。デフォルトでは、指定された仮想ディスクにスナップショットが含まれていると、破棄操作が失敗します。このサブコマンドには、次に示すオプションがあります。

`-r`
: 仮想ディスクを再帰的に破棄します。仮想ディスクに関連付けられたスナップショットもすべて破棄されます。

vdiskadm snapshot

`vdiskadm snapshot` *vdname@snapname*

仮想ディスクのスナップショットを、指定された *snapname* で作成します。このサブコマンドにオプションはありません。

vdiskadm rollback

`vdiskadm rollback [-r]` *snapshot*

仮想ディスクを以前のスナップショットにロールバックします。仮想ディスクをロールバックすると、スナップショット作成時から変更されたすべてのデータは破棄され、仮想ディスクがスナップショット作成時の状態に戻ります。デフォルトでは、このコマンドを使って、最新のスナップショット以外のスナップショットにロールバックすることはできません。最新でないスナップショットにロールバックする場合は、`-r` オプションを使って中間スナップショットをすべて破棄する必要があります。このサブコマンドには、次に示すオプションがあります。

`-r`
指定したスナップショット以降のスナップショットをすべて再帰的に破棄します。

vdiskadm clone

`vdiskadm [-c` *comment*`]` *vdname*`|`*snapshot* *clone_vdname*

指定されたスナップショットまたは仮想化ディスクのクローンを作成します。クローンの作成時には、クローン元となる仮想ディスクのタイプ、オプション、およびサイズが使用されます。*clone_vdname* にパスが含まれている場合、そのパスに付随するディレクトリが、クローン仮想ディスクの作成時に作成されます。デフォルトではマージクローンイメージが作成されます。このサブコマンドには、次に示すオプションがあります。

`-c` *comment*
クローン仮想ディスクに添付可能なコメント。

vdiskadm move

`vdiskadm move` *vdname* *dir*

指定された仮想ディスクを、指定されたディレクトリに移動します。仮想ディスクの名前はそのまま維持されます。新しいディレクトリは存在する必要があります。このサブコマンドにオプションはありません。

vdiskadm rename

`vdiskadm rename` *vdname*`|`*snapshot* *vdname*`|`*snapshot*

仮想ディスクまたはスナップショットの名前を変更します。このサブコマンドにオプションはありません。

vdiskadm list

`vdiskadm list [-fp]`*vdname*

指定された仮想ディスクとそのスナップショットを一覧表示します。このサブコマンドには、次に示すオプションがあります。

`-f`
仮想ディスクに関連するすべてのファイルのリストを出力します。このリストにはストアファイルやエクステントが含まれます。

-p
容易に解析可能な形式でファイルを一覧表示します。各ファイルの先頭にはラベル file:、snapshot:、または store が付けられます。

vdiskadm verify

vdiskadm verify *vdname*

Solaris が仮想ディスクを認識または開くことができない場合にエラーを返します。このサブコマンドにオプションはありません。

vdiskadm prop-get

vdiskadm prop-get [-l] -p *property* *vdname*

指定された仮想ディスクのプロパティーの値を返します。プロパティー値 all を使用すると、仮想ディスクのすべてのネイティブプロパティーとユーザー定義プロパティーが表示されます。このサブコマンドには、次に示すオプションがあります。

-l
プロパティーの書き込み可能ステータスなど、プロパティーの追加情報を出力します。

-p *property*
クエリー対象のプロパティーを指定し、そのプロパティーの値を表示します。プロパティー all の場合、プロパティーの値の前にプロパティーの名前、コロン、およびスペースが表示されます。

vdiskadm prop-set

vdiskadm prop-set -p *property=value* *vdname*

指定された仮想ディスクの指定されたプロパティーの値を設定します。*property* はネイティブプロパティー、ユーザー定義プロパティーのどちらでもかまいませんが、書き込み可能である必要があります。prop-add サブコマンドで追加されたプロパティーの値を変更するために使用できます。このサブコマンドには、次に示すオプションがあります。

-p *property=value*
設定するプロパティーを指定します。

vdiskadm prop-add

vdiskadm -p *property=value* *vdname*

指定された値を持つユーザー定義プロパティーを、指定された仮想ディスクに追加します。そのプロパティーがすでに存在している場合はエラーを返します。ユーザー定義プロパティーの名前には、コロン文字 (:) を含める必要があります。このサブコマンドには、次に示すオプションがあります。

-p *property=value*
追加するプロパティーを指定します。

vdiskadm prop-del

vdiskadm prop-del -p *property* *vdname*

指定された仮想ディスクからユーザー定義プロパティーを削除します。このサブコマンドには、次に示すオプションがあります。

-p *property*
削除するプロパティーを指定します。

vdiskadm import

```
vdiskadm import [-fnpqm] [-x type] -d file|zvol|dsk \
[-t type[:opt]] vdname
```

ファイルまたはブロック型デバイスのデータを使用して新しい仮想ディスクを作成します。ファイルの形式は vdi、vhd、vmdk、raw のいずれかになります。ブロック型デバイスは常に raw 形式であるとみなされます。このサブコマンドには、次のオプションがあります。

-f
インポート処理で使用されるファイルのリストを返します。

-n
インポートを実際に実行せずにインポートの出力を表示します。

-p
ファイルを容易に解析可能な形式で表示します。

-q
非出力モードで実行し、出力を一切表示しません。

-m
データのコピーは行わずにインポート対象ファイルを仮想ディスクに移動します。

-x *type*
インポートする仮想ディスクデータのタイプを指定します。インポート対象ファイルのタイプを vdiskadm が検出できない場合は、-x を指定する必要があります。

-d file|zvol|dsk
インポートするデータを含むファイルまたはブロック型デバイス。

-t *type*[:*opt*]
インポート時に作成される仮想ディスクのタイプを指定します。デフォルトのタイプは vmdk です。vmdk、vdi、および vhd タイプの場合、デフォルトの *opt* は sparse になります。タイプ raw の場合、デフォルトの *opt* は fixed になります。

vdiskadm export

```
vdiskadm export -x type[:opt] -d file|zvol|dsk vdname
```

仮想ディスクのデータをファイルまたはブロック型デバイスにエクスポートします。このサブコマンドには、次のオプションがあります。

-x *type*[:*opt*]
エクスポートする仮想ディスクデータのタイプを指定します。

-d file|zvol|dsk
エクスポートされたデータを受け取るファイルまたはブロック型デバイス。

vdiskadm convert

`vdiskadm convert [-t type[:opt]] vdname`

仮想ディスクを別のタイプの仮想ディスクに変換します。このサブコマンドには、次のオプションがあります。

`-t` *type*[:*opt*]
変換時に作成される仮想ディスクのタイプを指定します。デフォルトのタイプは `vmdk` です。`vmdk`、`vdi`、および `vhd` タイプの場合、デフォルトの *opt* は `sparse` になります。タイプ `raw` の場合、デフォルトの *opt* は `fixed` になります。

vdiskadm translate

`vdiskadm translate [-i type[:opt]] -I input_file \`
`-x type[:opt] -d output_file`

仮想ディスクを作成せずに、ある仮想ディスクデータタイプから別のタイプにデータを変換します。このサブコマンドには、次のオプションがあります。

`-i` *type*[:*opt*]
変換する仮想ディスクデータの入力タイプを指定します。入力ファイルのタイプを `vdiskadm` が検出できない場合は、`-i` を指定する必要があります。

`-I` *input_file*
変換するデータを含むファイルまたはブロック型デバイス。

`-x` *type*[:*opt*]
変換する仮想ディスクデータの出力タイプを指定します。`vmdk`、`vdi`、および `vhd` タイプの場合、デフォルトの *opt* は `sparse` になります。タイプ `raw` の場合、デフォルトの *opt* は `fixed` になります。

`-d` *output_file*
変換されたデータを受け取るファイルまたはブロック型デバイス。*output_file* は存在してはいけません。このファイルは変換中に作成されます。

vdiskadm help

`vdiskadm help [command]`

一般的なヘルプメッセージまたはコマンド固有のヘルプメッセージを表示します。このサブコマンドの引数は、省略可能なコマンド名だけです。

使用例

例 1　vmdk 疎ファイルを作成する

次のコマンドは、サイズが 8G バイトの `disk1` という名前の仮想ディスクを、ディレクトリ `/guests/disks` に作成します。

```
# vdiskadm create -s 8g -t vmdk:sparse /guests/disks/disk1
```

例 2　スナップショットを作成する

次のコマンドは、`/guests/disks/disk1` にある仮想ディスクのスナップショットを作成します。スナップショットの名前は `install` です。

```
# vdiskadm snapshot /guests/disks/disk1@install
```

例 3 スナップショットを作成および破棄する

次のコマンドは、/guests/disks/disk1 にある仮想ディスクの、install、bfu という名前の 2 つのスナップショットを作成します。3 つ目のコマンドは、新しく作成されたスナップショット install を破棄します。

```
# vdiskadm snapshot /guests/disks/disk1@install
# vdiskadm snapshot /guests/disks/disk1@bfu
# vdiskadm destroy /guests/disks/disk1@install
```

例 4 仮想ディスクをロールバックする

次のコマンドは、仮想ディスクの内容を install という名前のスナップショットに戻し、中間のスナップショットをすべて削除します。

```
# vdiskadm rollback -r /guests/disks/disk1@install
```

例 5 仮想ディスクとスナップショットを一覧表示する

次のコマンドは、仮想ディスク /guests/disks/disk1 に関連付けられたすべてのイメージを一覧表示します。

```
# vdiskadm list /guests/disks/disk1
disk1@install
disk1@bfu
disk1
```

例 6 クローンを作成する

次のコマンドは、仮想ディスク /guests/disks/disk1 の合体したコピーとなる仮想ディスクを新しく作成します。このクローンは、元の仮想ディスクと同じ形式 (つまり vmdk:sparse) で作成されます。

```
# vdiskadm clone /guests/disks/disk1 /guests/clone/clone_disk1
```

例 7 ユーザー定義プロパティーを追加する

次のコマンドは、仮想ディスクにユーザー定義プロパティーを追加し、指定された値をそのプロパティーに設定します。このプロパティー名は、この仮想ディスクデータの提供元と要件を表すために選択されたものであり、フィールドを示すために必須のコロンが使用されています。

```
# vdiskadm prop-add -p com.sun:required-nic=2 /guests/disks/disk1
```

例 8 既存の vmdk 形式ファイルをインポートする

次のコマンドは、既存の vmdk 形式ファイルを受け取り、それを仮想ディスクにインポートします。

```
# vdiskadm import -d /downloads/appliance.vmdk /guests/import/disk1
```

例 9 vmdk ファイルをインポートして vhd に変換する

次のコマンドは、既存の vmdk 形式ファイルを受け取り、インポート時にそれを vhd タイプの仮想ディスクに変換します。

```
# vdiskadm import -d /downloads/appliance.vmdk -t vhd \
/guests/import/disk1
```

例 10 zvol からデータをインポートする

次のコマンドは、zvol から仮想ディスクデータをインポートし、インポート時にそれを vmdk:fixed タイプの仮想ディスクに変換します。

```
# vdiskadm import -d /dev/zvol/dsk/pool1/disk1 -t vmdk:fixed \
/guests/import/disk1
```

例 11 ブロック型デバイスにデータをエクスポートする

次のコマンドは、既存の仮想ディスクを受け取り、エクスポート時にそれを raw タイプのディスクスライスに変換します。

```
# vdiskadm export -d /dev/dsk/c0t1d0s3 -x raw /guests/disks/disk1
```

例 12 仮想ディスクのタイプを変換する

次のコマンドは、既存の仮想ディスクを受け取り、それを (その場で) 別の形式タイプに変換します。

```
# vdiskadm convert -t vdi:fixed /guests/disks/disk1
```

例 13 データをあるタイプから別のタイプに変換する

次のコマンドは、仮想ディスクを作成せずに、ある仮想ディスク形式のファイルのデータを raw データに変換して zvol に書き込みます。

```
# vdiskadm translate -I /downloads/appliance.vmdk -x raw \
-d /dev/zvol/dsk/pool1/disk1
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/storage/vdiskadm
インタフェースの安定性	不確実

関連項目

attributes(5)

名前

virt-convert – convert virtual machines between formats

形式

```
virt-convert [option]... input.vmx | input.ovf | input-dir
      [output.xml | output-dir]
```

機能説明

The `virt-convert` program is a command line tool for converting virtual machines (VMs) from one format to another. It accepts either a VM definition file (such as VMware `vmx` format) or a directory containing a VM. By default, a new VM definition file, and converted disk images, will be placed in a new output directory.

If an output directory is specified, it will be created if necessary, and the output VM definition placed within the new directory, along with any disk images, as needed.

If an output VM definition file is specified, it will be created alongside any disks in the same directory.

オプション

Any of the following options can be omitted, in which case `virt-convert` will use defaults when required. An input VM definition or containing directory must be provided. By default, an output directory is generated based upon the name of the VM. The default input format is VMware `vmx`, and the default output format is a `libvirt` "image" XML definition.

`-a`, `--arch=`*arch*
: Architecture of the virtual machine (`i686`, `x86_64`, `ppc`). Defaults to that of the host machine.

`-D`, `--disk-format=`*format*
: Output disk format, or `none` if no conversion should be performed. *format* is one of:

  `none`
  : No disks are converted or copied.

  `vmdk`
  : VMWare VMDK format

  `raw`
  : raw file

  `vdisk`
  : `vdisk` format (see `vdiskadm(1M)`)

`-d`, `--debug`
: Display debugging information.

`-h`, `--help`
: Display the help message and exit.

`-i`, `--input-format` *format*
: Input format. Currently, `vmx` and `ovf` are supported.

`--noacpi`
: Override the OS type and variant to disable the ACPI setting for fully virtualized guest.

`--noapic`
Override the OS type and variant to disable the APIC setting for fully virtualized guest.

`-o`, `--output-format` *format*
Output format. Currently, the supported output formats are `virt-instance` and `virt-image`. `virt-instance` is the recommended format for Solaris.

`--os-type=`*os_type*
Optimize the guest configuration for a type of operating system. This will attempt to pick the most suitable ACPI and APIC settings, optimally supported mouse drivers, and generally accommodate other operating system quirks.

`--os-variant=`*os_variant*
Further optimize the guest configuration for a specific operating system variant. This parameter is optional.

`-p`, `--paravirt`
Create a paravirtualized guest image. Convert machine to a paravirtualized Xen-based image.

`-q`, `--quiet`
Avoid verbose output.

`-v`, `--hvm`
Create a fully virtualized guest image. Convert machine to a `hvm`/`qemu`–based image (this is the default if `--paravirt` is not specified).

使用例

例 1 Converting a VMware VMX appliance

The following sequence of commands converts a VMware VMX appliance and imports it into Solaris xVM.

```
# virt-convert -o virt-instance /guests/vmx-appliance/ \
/guests/xvm-appliance/
# virsh define --relative-path=/guests/xvm-appliance/ \
\guests/xvm-appliance/appliance.xml
```

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xvm/header-xvm
Interface Stability	Volatile

関連項目

`vdiskadm(1M)`, `attributes(5)`

Caveats

Not all conversions will result in a working guest installation. If the source OS image is configured to use SCSI disks, the use of IDE disks may cause the OS boot to fail. Some images may be configured to use the VMware drivers such as `vlance`. In the `vlance` case and in general, device emulation support may not be sufficient for all OS installations.

Authors

Written by Joey Boggs and John Levon.

See the `AUTHORS` file in the source distribution for the complete list of credits.

名前

virtinfo – virtualization domain information

形式

```
virtinfo [ -acdpstu ]
```

機能説明

The `virtinfo` utility provides virtualization information returned by `libv12n(3LIB)` about the current domain. When options are specified, symbols that represent one or more virtualization domain characteristics are written to standard output. If no options are specified, `virtinfo` shows only the virtualization domain type.

オプション

The `virtinfo` command supports the following options:

`-a` Shows all information that is currently available from the system.

`-c` Shows the network node name of the control domain. This name is identical to the string shown by running the `uname -n` command on the control domain.

`-d` Shows the domain name for this virtual domain.

`-p` Shows information in a parseable format.

The parseable output for Logical Domains (LDoms) has the following format:

```
VERSION 1.0
DOMAINROLE|impl=LDoms|control={true|false}|
    io={true|false}|service={true|false}|
    root={true|false}
DOMAINNAME|name=domain-name
DOMAINUUID|uuid=uuid
DOMAINCONTROL|name=control-domain-nodename
DOMAINCHASSIS|serialno=serial-no
```

`-s` Shows the platform serial number.

`-t` Shows the domain type.

The domain type for Logical Domains is a string of one or more of the following blank-separated values:

`LDoms`	LDoms implementation
`control`	Control domain
`guest`	Guest domain
`I/O`	I/O domain
`service`	Service domain
`root`	Root I/O domain

Note that the `control` and `guest` values are mutually exclusive.

-u　Shows the universally unique identifier (UUID) of the domain. See the libuuid(3LIB) man page.

終了ステータス

The following exit values are returned:

0　Successful completion.

>0　An error occurred.

使用例

例 1　Viewing Default Information

The following example shows the default information about the domain:

```
$ virtinfo
Domain role: LDoms control I/O service root

The current domain is the control domain, which is also an
I/O domain, the service domain and a root I/O domain.
```

例 2　Viewing All Information

The following example shows all available information about the domain:

```
$ virtinfo -a
Domain role: LDoms control I/O service root
Domain name: primary
Domain UUID: 8e0d6ec5-cd55-e57f-ae9f-b4cc050999a4
Control domain: san-t2k-6
Chassis serial#: 0704RB0280
```

例 3　Viewing All Information In a Parsable Format

The following example shows all available information about the domain in a parsable format:

```
$ virtinfo -ap
VERSION 1.0
DOMAINROLE|impl=LDoms|control=true|io=true|service=true|root=true
DOMAINNAME|name=primary
DOMAINUUID|uuid=8e0d6ec5-cd55-e57f-ae9f-b4cc050999a4
DOMAINCONTROL|name=san-t2k-6
DOMAINCHASSIS|serialno=0704RB0280
```

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

関連項目 libuuid(3LIB), libv12n(3LIB)

名前

vmstat – 仮想メモリーの統計情報の報告

形式

`vmstat [-ipqsS] [-T u | d] [disks] [interval [count]]`

機能説明

vmstat は、カーネルスレッド、仮想メモリー、ディスク、トラップ、および CPU アクティビティーに関連した仮想メモリーの統計情報を報告します。

MP (マルチプロセッサ) システムでは、vmstat は CPU の数を平均して出力します。プロセッサごとの統計情報については、mpstat(1M) を参照してください。

vmstat は、特定のデバイスの統計情報のみをサポートしています。より一般的なシステム統計情報を取得するには、sar(1)、iostat(1M)、または sar(1M) を使用してください。

オプションを指定しないと、vmstat は、システムがブートされてからの仮想メモリーアクティビティーの要約を 1 行で表示します。

カーネルステータスコマンドの実行中に、システムのステータスが変更されることがあります。関連する場合は、次のいずれかの形式の状態変更メッセージが vmstat の出力に含まれます。

```
<<device added: sd0>>
<<device removed: sd0>>
<<processors added: 1, 3>>
<<processors removed: 1, 3>>
```

ディスクのデバイス命名規則については、『『Oracle Solaris の管理: 一般的なタスク』』を参照してください。

オプション

次のオプションがサポートされています。

`-i` デバイスごとの割り込みの数を報告します。*count* と *interval* は、-i オプションには適用されません。

`-p` 詳細なページングアクティビティーを報告します。このオプションでは、それぞれ次の情報が表示されます。

- epi 実行可能ファイルのページイン。
- epo 実行可能ファイルのページアウト。
- epf 実行可能ファイルのページフリー。
- api 匿名のページイン。
- apo 匿名のページアウト。
- apf 匿名のページフリー。
- fpi ファイルシステムのページイン。
- fpo ファイルシステムのページアウト。
- fpf ファイルシステムのページフリー。

zone で実行中にプール機能がアクティブな場合は、上のすべてで、その zone のプールのプロセッサセット内のプロセッサに関するアクティビティーのみが報告されます。

-q　状態変更に関連したメッセージを抑制します。

-s　ブート時からのさまざまなシステムイベントの総数を表示します。*count* と *interval* は、-s オプションには適用されません。

-S　ページングアクティビティーではなく、スワッピングに関して報告します。このオプションは、vmstat の “ページング" 表示内の 2 つのフィールドを変更します。“re" および “mf" フィールドではなく、vmstat は “si" (スワップイン) と “so" (スワップアウト) を報告します。

-T u|d　時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。 標準の日付フォーマットに d を指定します。date(1) を参照してください。

オペランド

次のオペランドがサポートされています。

count　統計を繰り返す回数を指定します。*count* は、-i および -s オプションには適用されません。

disks　出力で優先順位を与えるディスクを指定します (1 行に収まるのは 4 つのディスクだけです)。一般的なディスク名は、id、sd、xd、または xy のあとに数字が付きます (たとえば、sd2、xd0 など)。

interval　vmstat がアクティビティーを集計する最後の秒数を指定します。この秒数は永久に繰り返されます。*interval* は、-i および -s オプションには適用されません。

使用例

例 1　vmstat の使用

次のコマンドは、システムで実行されている処理の要約を 5 秒ごとに表示します。

```
example% vmstat 5


kthr   memory          page             disk      faults        cpu
r b w swap  free re mf pi p fr de sr s0 s1 s2 s3  in  sy  cs us sy id
0 0 0 11456 4120 1  41 19 1  3  0  2  0  4  0  0  48 112 130  4 14 82
0 0 1 10132 4280 0   4 44 0  0  0  0  0 23  0  0 211 230 144  3 35 62
0 0 1 10132 4616 0   0 20 0  0  0  0  0 19  0  0 150 172 146  3 33 64
0 0 1 10132 5292 0   0  9 0  0  0  0  0 21  0  0 165 105 130  1 21 78
1 1 1 10132 5496 0   0  5 0  0  0  0  0 23  0  0 183  92 134  1 20 79
1 0 1 10132 5564 0   0 25 0  0  0  0  0 18  0  0 131 231 116  4 34 62
1 0 1 10124 5412 0   0 37 0  0  0  0  0 22  0  0 166 179 118  1 33 67
1 0 1 10124 5236 0   0 24 0  0  0  0  0 14  0  0 109 243 113  4 56 39
^C
```

例 1　vmstat の使用　　*(続き)*

```
example%
```

vmstat の表示の各フィールドは次のとおりです。

kthr　次の 3 つの各状態にあるカーネルスレッドの数を報告します。

- r　実行待ち行列にあるカーネルスレッドの数
- b　リソース I/O、ページングなどを待機中のブロックされたカーネルスレッドの数
- w　リソース処理の完了を待機中のスワップアウトされた軽量プロセス (LWP) の数

memory　仮想メモリーと実メモリーの使用状況に関して報告します。

- swap　使用可能なスワップ空間 (K バイト)
- free　空きリストのサイズ (K バイト)

page　ページフォルトおよびページングアクティビティーに関する情報を報告します。次の各アクティビティーに関する情報が 1 秒あたりの単位で表示されます。

- re　ページの再利用。ただし、このフィールドの変更方法については -S オプションを参照してください。
- mf　マイナーフォルト。ただし、このフィールドの変更方法については -S オプションを参照してください。
- pi　ページインされた K バイト数
- po　ページアウトされた K バイト数
- fr　解放された K バイト数
- de　予測される短期間のメモリー不足 (K バイト)
- sr　クロックアルゴリズムによってスキャンされたページ数

zone で実行中にプール機能がアクティブな場合は、上のすべて ("de" を除く) で、その zone のプールのプロセッサセット内のプロセッサに関するアクティビティーのみが報告されます。

disk　1 秒あたりのディスク操作の数を報告します。1 文字と数字で示された、最大 4 つのディスクのスロットがあります。この文字はディスクのタイプ (s = SCSI、i = IPI など) を示し、数字は論理ユニット番号です。

faults　トラップ/割り込み率 (1 秒あたり) を報告します。

例 1 vmstat の使用 (続き)

in 割り込み

sy システムコール

cs CPU コンテキストスイッチ

zone で実行中にプール機能がアクティブな場合は、上のすべてで、その zone のプールのプロセッサセット内のプロセッサに関するアクティビティーのみが報告されます。

cpu CPU 時間のパーセント使用状況の内訳を表示します。MP システムでは、これはすべてのプロセッサにわたる平均です。

us ユーザー時間

sy システム時間

id アイドル時間

zone で実行中にプール機能がアクティブな場合は、上のすべてで、その zone のプールのプロセッサセット内のプロセッサに関するアクティビティーのみが報告されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	下記を参照

この呼び出しは開発中です。人間が読める形式の出力は不安定 (Unstable) です。

関連項目

date(1), sar(1), iostat(1M), mpstat(1M), sar(1M), time(2), attributes(5)

注意事項

CPU 使用率の合計は、100 からわずかにずれることがあります。これは、割合を示す数字を生成するときの四捨五入エラーのためです。

名前

vntsd – 論理ドメイン用の仮想ネットワーク端末サーバーデーモン

形式

/usr/lib/ldoms/vntsd

機能説明

vntsd デーモンは、telnet(1) を使用した論理ドメイン (LDoms) コンソールへの接続をサポートするサーバーです。telnet セッションが開始すると、vntsd は、リモートで文字をエコーする要求と Go Ahead を抑制する要求を示す telnet オプションを、クライアントに送信します。

コンソールは、LDoms Manager によってグループにまとめられます。各コンソールグループには一意のグループ名と TCP ポート番号が割り当てられます。vntsd は、グループのポート番号を使用してそのグループ内のコンソールへのアクセスをエクスポートします。ユーザーは、コンソールまたはコンソールグループとの接続を確立するために、対応するグループのポート番号を使用して telnet(1) セッションを開始します。そのグループ内のコンソールの数に応じて、vntsd は次の 2 つの処理のいずれかを行います。

- グループ内にコンソールが 1 つしか存在しない場合、vntsd はその LDoms コンソールにセッションを接続します。
- グループ内にコンソールが複数存在する場合、vntsd は、接続したいコンソールを選択するようユーザーに促します。詳細は、後述の「複数コンソールオプション」を参照してください。

各コンソールについて、vntsd はコンソールに最初に接続したユーザーだけに書き込みアクセス権を付与します。その後にコンソールに接続したユーザーは、コンソールからの読み取りしか許可されず、書き込みアクセスについては待機状態となります。最初のユーザーの接続が切断されると、キュー内で待機している次のユーザーに書き込み特権が移されます。書き込み特権を持たないユーザーがコンソールへの書き込みを試みると、vntsd によって次のメッセージが表示されます。

```
You do not have write access
```

書き込みアクセス権を持たないユーザーが書き込みアクセス権を強制的に取得するには、特殊コンソールコマンド ~w を使用します。詳細は、後述の「特殊コンソールコマンド」を参照してください。

vntsd の呼び出しは、スーパーユーザー特権を使用して行うか、Primary Administrator 役割のユーザーが行う必要があります。

オプション

vntsd のオプションは、複数コンソールオプションとコンソールコマンドに分けられます。

複数コンソールオプション

グループ内に LDoms コンソールが複数存在している場合、次に示すオプションがサポートされます。これらのオプションを使用するための構文は次のとおりです。

<hostname>-vnts-*<group-name>*: *<option>*

例:

```
myhost-vnts-salesgroup: h
```

h オプションは、次に説明するようにヘルプを呼び出します。

h
: 次のヘルプテキストを表示します。

```
h -- this help
l -- list of consoles
q -- quit
c{id}, n{name} -- connect to console of domain {id} or domain name
```

l
: グループ内のすべてのコンソールを一覧表示します。例:

```
DOMAIN ID       DOMAIN NAME     DOMAIN STATE
0               ldg1            online
1               ldg2            connected
...             ...             ...
```

2 つのドメイン状態とその意味を次に示します。

online
: コンソールに接続しているユーザーが一人もいません。

connected
: 少なくとも一人のユーザーがすでにコンソールに接続しています。

q
: vntsd への接続を切断します。

c*{id}*、n*{name}*
: 指定されたコンソールに接続します。接続時に次のメッセージが表示されます。

```
Connecting to console <domain-name> in group <group-name>
Press ~? for control options ....
```

特殊コンソールコマンド

行の最初にあるチルド (~) はエスケープ信号で、vntsd に特殊コンソールコマンドを実行させます。連続するチルド (~~) を指定すると、チルドが出力されます。vntsd は、先頭のチルドとともに次の特殊コンソールコマンドを受け入れます。

~.
: コンソールまたはコンソールグループへの接続を切断します。

~w
: コンソールへの書き込みアクセスを強制的に行います。

~p
: このコンソールへの接続を切断し、コンソールリスト内でこのコンソールの前にあるコンソールに接続します。

~n
このコンソールへの接続を切断し、コンソールリスト内でこのコンソールの後にあるコンソールに接続します。

~#
ブレークを送信します。

~^B
代替ブレークを送信します。

~?
次のような vntsd のヘルプを表示します。

```
~# - Send break
~^B - Send alternate break
~. - Exit from this console
~w - Force write access
~n - Console next
~p - Console previous
~? - Help
```

ファイル

/usr/lib/ldoms/vntsd
vntsd のバイナリ実行可能ファイル。

/usr/lib/ldoms/vntsd.xml
vntsd 用のサービス管理機能 (smf(5)) マニフェストファイル。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/ldoms
インタフェースの安定性	確実

関連項目

telnet(1), svccfg(1M), usermod(1M), auth_attr(4), attributes(5), smf(5)

注意事項

vntsd はサービス管理機能 smf(5) により次のサービス識別子の下で管理されます。

```
svc:/ldoms/vntsd
```

svccfg(1M) コマンドを使用して次のプロパティーを変更できます。

vntsd/vcc_device
vntsd の接続先となる仮想コンソール端末集配信装置 (vcc) ドライバのインスタンスを設定します。

vntsd/listen_addr
vntsd が待機する IP アドレスを、次の構文を使って設定します。

```
vntsd/listen_addr:"xxx.xxx.xxx.xxx"
```

ここで、*xxx.xxx.xxx.xxx* は有効な IP アドレスです。このプロパティーのデフォルト値は、IP アドレス 127.0.0.1 上で待機します。この値を制御ドメインの IP アドレスに設定すると、ユーザーがネットワーク経由でゲストコンソールに接続できるようになります。

注 – コンソールへのネットワークアクセスを有効にすると、セキュリティーの問題が発生します。任意のユーザーがコンソールに接続できます。このため、この機能はデフォルトで無効化されます。

`vntsd/timeout_minutes`
タイムアウトを分単位で設定します。コンソールのアクティビティー (入力または出力) がまったく発生しないと、`vntsd` は `telnet` 接続をタイムアウトします (閉じます)。デフォルト値は 0 で、この場合はタイムアウトが無効化されます。

`vntsd/authorization`
アクセス中の 1 つまたは複数のドメインコンソールに対してユーザーや役割の承認チェックを有効にします。このプロパティーのデフォルト値は `false` ですが、これは、下位互換性を維持するためです。承認チェックを有効にするには、`svccfg(1M)` コマンドを使用してプロパティー値を `true` に設定します。このオプションが有効化されていると、`vntsd` は、`localhost` 上で接続を待機して受け入れます。このオプションが有効化された状態で `listen_addr` プロパティーに代替 IP アドレスが指定されても、`vntsd` はその代替 IP アドレスを無視し、`localhost` 上で待機し続けます。ほかのホストから開始された接続も失敗します。すべてのコンソールまたはコンソールグループにアクセスする承認と、特定のコンソールまたはコンソールグループにアクセスする承認が使用できます。`vntsd` サービスを有効化すると、承認記述データベース `auth_attr(4)` に次の承認が追加されます。

```
solaris.vntsd.consoles:::Access All LDoms Guest Consoles::
```

コンソールグループの名前に基づく、きめ細かな承認を追加します。たとえば、承認対象となるコンソールグループの名前が `ldg1` の場合、`auth_attr(4)` ファイルに次のエントリを追加します。

```
solaris.vntsd.console-ldg1:::Access Specific LDoms Guest Console::
```

ルートユーザーまたはルート役割にはデフォルトで、すべてのコンソールにアクセスする承認が割り当てられます。Primary Administrator (スーパーユーザー) は、`usermod(1M)` コマンドを使用すると 1 つまたは複数の必要な承認をほかのユーザーや役割に割り当てることができます。

次の例では、ユーザー `user1` に、すべてのドメインコンソールにアクセスする承認を付与しています。

```
# usermod -A "solaris.vntsd.consoles" user1
```

次の例では、ユーザー `user1` に、`ldg1` という名前のコンソールグループにアクセスする承認を付与しています。

```
# usermod -A "solaris.vntsd.console-ldg1" user1
```

名前

volcopy – make an image copy of file system

形式

```
volcopy [-F FSType] [-V] [generic_options]
      [-o FSType-specific_options] operands
```

機能説明

`volcopy` makes a literal copy of the file system. This command may not be supported for all *FSTypes*.

オプション

The following options are supported:

`-F` *FSType*
: Specify the *FSType* on which to operate. The *FSType* should either be specified here or be determinable from `/etc/vfstab` by matching the *operands* with an entry in the table. Otherwise, the default file system type specified in `/etc/default/fs` will be used.

`-V`
: Echo the complete command line, but do not execute the command. The command line is generated by using the options and arguments provided by the user and adding to them information derived from `/etc/vfstab`. This option should be used to verify and validate the command line.

generic_options
: Options that are commonly supported by most *FSType*-specific command modules. The following options are available:

  `-a` Require the operator to respond “yes” or “no” instead of simply waiting ten seconds before the copy is made.

  `-s` (Default) Invoke the DEL if wrong verification sequence.

`-o` *FSType-specific_options*
: Specify *FSType*-specific options in a comma separated (without spaces) list of suboptions and keyword-attribute pairs for interpretation by the *FSType*-specific module of the command.

オペランド

The following operands are supported:

operands generally include the device and volume names and are file system specific. A detailed description of the *operands* can be found on the *FSType*-specific man pages of `volcopy`.

終了ステータス

The following exit values are returned:

`0` Successful file system copy

`1` An error has occurred.

ファイル

`/etc/vfstab` list of default parameters for each file system

`/etc/default/fs` default local file system type. Default values can be set for the following flags in `/etc/default/fs`. For example: `LOCAL=ufs`.

`LOCAL:` The default partition for a command if no *FSType* is specified.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`labelit(1M)`, `vfstab(4)`, `attributes(5)` Manual pages for the FSType-specific modules of `volcopy`.

名前

volcopy_ufs – make an image copy of a ufs file system

形式

`volcopy [-F ufs] [`*generic_options*`] `*fsname srcdevice volname1 destdevice volname2*

機能説明

`volcopy` makes a literal copy of the `ufs` file system using a blocksize matched to the device.

オプション

The following option is supported:

generic_options options supported by the generic `volcopy` command. See `volcopy(1M)`.

オペランド

The following operands are supported:

fsname represents the mount point (for example, `root`, `u1`, etc.) of the file system being copied.

srcdevice or *destdevice* the disk partition specified using the raw device (for example, `/dev/rdsk/cld0s8`, `/dev/rdsk/cld1s8`, etc.).

srcdevice and *volname1* the device and physical volume from which the copy of the file system is being extracted.

destdevice and *volname2* the target device and physical volume.

fsname and *volname* are limited to six or fewer characters and recorded in the superblock. *volname* may be '–' to use the existing volume name.

終了ステータス

The following exit values are returned:

`0` Successful file system copy.

non-zero An error has occurred.

ファイル

`/var/adm/filesave.log` a record of file systems/volumes copied

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`cpio(1)`, `dd(1M)`, `labelit(1M)`, `volcopy(1M)`, `attributes(5)`, `ufs(7FS)`

注意事項

`volcopy` does not support copying to tape devices. Use `dd(1M)` for copying to and from tape devices.

名前

vrrpadm – VRRP 管理ツール

形式

```
vrrpadm create-router -V vrid -l link -A inet | inet6
        [-p priority] [-i adv_interval] [-o flags] router_name

vrrpadm delete-router router_name

vrrpadm disable-router router_name

vrrpadm enable-router router_name

vrrpadm modify-router [-p priority] [-i adv_interval]
        [-o flags] [router_name]

vrrpadm show-router [-P | -x] [-p] [-o field[,...]] [router_name]
```

機能説明

`vrrpadm` コマンドは、システム内の VRRP (Virtual Router Redundancy Protocol) サービスを管理するために使用します。

VRRP は選択プロトコルを指定し、この選択プロトコルによって、仮想ルーターの役割が LAN 内のいずれかの VRRP ルーターに動的に割り当てられます。任意の時点で、仮想ルーターに関連付けられた IPv4 または IPv6 仮想アドレスを制御する VRRP ルーター (「マスター」と呼ばれる) は 1 つだけであり、マスターはそれらの IP アドレスに送信されたパケットを転送します。マスターが使用不可能になった場合に、選択処理によって転送の役割が動的にフェイルオーバーされます。

各 `vrrpadm` サブコマンドの処理対象は VRRP ルーターです。VRRP ルーターは管理者から与えられた名前で識別されます。1 つの LAN 内で同じ VRID とアドレスファミリを持つ複数の VRRP ルーターによって 1 つの仮想ルーターが構成されます。仮想ルーターは一連の仮想 IP アドレスを保護します。

システム上に複数の VRRP ルーターが存在し、各ルーターがそれぞれ異なる仮想ルーターに所属できます。

サブコマンド

次のサブコマンドがサポートされています。`show-router` 以外のすべてのサブコマンドでは、`solaris.network.vrrp` 承認が必要となります。`show-router` サブコマンドに特別な承認は不要です。

`vrrpadm create-router -V` *vrid* `-l` *link* `-A` *inet* | *inet6* [`-p` *priority*] [`-i` *adv_interval*] [`-o` *flags*] *router_name*

指定された構成で VRRP ルーターを作成します。

`-A` *inet* | *inet6*、`--address_family=`*inet* | *inet6*

アドレスファミリ。IPv4、IPv6 のいずれか。

`-i` *adv_interval*、`--adv_interval=`*adv_interval*

通知間隔 (ミリ秒)。デフォルトは `1000` (1 秒) です。有効な間隔は、10 から 40950 の範囲です。

-l *link*、--link=*link*
VRRP ルーターの構成先となるデータリンク。これにより、この VRRP ルーターが動作する LAN が決まります。データリンクは物理リンク、VLAN、集約のいずれかになります。

-o *flags*、--flags=*flags*
横取りモードと受け入れモードをコンマで区切ったもの。使用可能な値は次のとおりです。

- preempt
- un_preempt
- accept
- noaccept

デフォルトでは、どちらのモードも真に設定されます。

preempt モードは、優先度の高い有効なバックアップルーターが優先度の低いマスタールーターを横取りするかどうかを制御します。preempt モードが真の場合は横取りが許可されます。それ以外の場合は横取りが禁止されます。VRRP ルーターが仮想 IP アドレスの所有者である場合には、preempt モードが真である必要があります。

accept モードは、仮想 IP アドレスのローカルパケットの受け入れを制御します。accept モードが真の場合、マスターは仮想 IP アドレスに送信されたパケットを受け入れる必要があります。accept モードが偽の場合、マスターはそれらのパケットを受け入れませんが、それらの受け入れない仮想 IP アドレスに対する ARP 要求や ND 要請/通知には応答します。さらにマスターは、このサブコマンドで指定されたルーターのパケットも転送する必要があります。VRRP ルーターが仮想 IP アドレスの所有者である場合には、accept モードが真である必要があります。このオプションの構文の例を次に示します。

```
-o preempt,no_accept
```

-p *priority*、--priority=*priority*
指定された VRRP ルーターの、マスター選択で使用される優先度。この値が高いほど、このルーターがマスターとして選択される可能性が高くなります。

デフォルト値は 255 です。この値は、指定された VRRP ルーターが IP アドレス所有者であり、すべての仮想 IP アドレスを所有していることを示します。IP アドレス所有者は、ICMP ping や TCP 接続などで発生する、いずれかの仮想 IP アドレス宛てのパケットに応答します。

1 から 254 の範囲は、仮想ルーターをバックアップする VRRP ルーターで使用できます。マスター選択では、優先度の高い VRRP ルーターが選択されるように重み付けされます。

`-V` *vrid*、`--VRID=`*vrid*
仮想ルーター識別子 (VRID)。これとアドレスファミリの組み合わせによって、LAN 内の 1 つの仮想ルーターが識別されます。

router_name
VRRP ルーターの名前。この名前は、ほかの `vrrpadm` サブコマンドで VRRP ルーターを識別するために使用されます。

有効なルーター名の最大長は、31 文字です。使用可能な文字は、英数字 (a-z、A-Z、0-9) と下線 (「_」) です。

`vrrpadm delete-router` *router_name*
router_name で識別される VRRP ルーターを削除します。

`vrrpadm disable-router` *router_name*
router_name で識別される仮想ルーターを無効にします。無効化されたルーターは、仮想ルーターのマスター選択処理の対象から外れます。

`vrrpadm enable-router` *router_name*
router_name で識別される無効化された仮想ルーターを再度有効にします。このルーターが再度、仮想ルーターのマスター選択処理の対象になります。

`vrrpadm modify-router` [`-p` *priority*] [`-i` *adv_interval*] [`-o` *flags*] [*router_name*]
router_name で識別される VRRP ルーターの構成を変更します。変更できるのは、優先度、通知間隔、横取りモード、および受け入れモードだけです。

`-p` *priority*、`--priority=`*priority*	この VRRP ルーターの新しい優先度。
`-i` *adv_interval*、`--adv_interval=`*adv_interval*	新しい通知間隔。
`-o` *flags*、`--flags=`*flags*	新しい横取りモードと受け入れモード。どちらか 1 つを指定することも、両方を指定することもできます。両方を指定する場合は、両者の間をコンマで区切ります。例: `-o preempt,no_accept`

`vrrpadm show-router` [`-P` | `-x`] [`-p`] [`-o` *field*[,...]] [*router_name*]
router_name で識別される VRRP ルーターの情報を表示します。*router_name* が指定されなかった場合は、システム上のすべての VRRP ルーターの情報を表示します。

デフォルトでは (オプションの指定なし)、次のフィールドが表示されます。

`NAME`
VRRP ルーターの名前。

VRID
VRRP ルーターの VRID。

LINK
VRRP ルーターの作成先となるデータリンク。

AF
VRRP ルーターのアドレスファミリ。IPv4 または IPv6。

PRIO
この VRRP ルーターの、マスター選択で使用される優先度。

ADV_INTV
通知間隔 (ミリ秒)。

STATE
VRRP ルーターの現在の状態。INIT (初期化)、BACK (バックアップ)、または MAST (マスター)。

MODE
VRRP ルーターに関連付けられたフラグのセット。表示される可能性のある値は次のとおりです。

e　ルーターが有効化されました。

p　横取りモードが真です。

a　受け入れモードが真です。

o　仮想アドレス所有者。

VNIC
この VRRP ルーター用に作成された VRRP VNIC。

ルーターが有効化されないかぎり、VNIC の名前は時間の経過とともに変化する可能性があります。

show-router サブコマンドのオプションは、次のとおりです。

-x、--extended
指定された VRRP ルーターの追加情報を表示します。

PRIMARY_IP
VRRP ルーターによって選択されたプライマリ IP アドレス。

VIRTUAL_IPS
VRRP ルーター上で構成されている仮想 IP アドレス。

PRV_STAT
VRRP ルーターの以前の状態。

STAT_LAST
前回の状態遷移からの時間。

-P、--peer
バックアップ VRRP ルーターの情報を表示します。このオプションが意味を持つのは、VRRP ルーターがバックアップ状態になっている場合だけです。

表示されるフィールドは次のとおりです。

NAME
VRRP ルーターの名前。

PEER
ピア VRRP ルーターのプライマリ IP アドレス。

P_PRIO
ピアから受信した通知に含まれていた、ピア VRRP ルーターの優先度。

P_INTV
ピアから受信した通知に含まれていた通知間隔 (ミリ秒)。

P_ADV_LAST
前回ピアから通知を受信してからの時間。

MASTER_DOWN_INT
マスター停止を宣言するまでの時間間隔 (ミリ秒)。

-p、--parseable
VRRP ルーターの情報を、マシン解析可能な形式で表示します。

-o *field*[,...]、--output=*field*
表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、前述のフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、vrrpadm show はすべてのフィールドを表示します。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/network/routing/vrrp
インタフェースの安定性	確実

関連項目

dladm(1M), vrrpd(1M), attributes(5)

名前

vrrpd – VRRP デーモン

形式

```
/usr/lib/vrrpd
```

```
svc:/network/vrrp:default
```

機能説明

vrrpd は、VRRP ルーターの管理イベントを処理するシステムデーモンプログラムです。このデーモンは、サービス管理機能 (SMF) サービスインスタンスによって制御されます。

```
svc:/network/vrrp:default
```

このデーモンを直接呼び出さないようにしてください。このデーモンは、管理インタフェースやプログラミングインタフェースの構成要素ではありません。VRRP ルーターを管理するための管理インタフェースは、vrrpadm(1M) です。

オプション

このデーモンにオプションはありません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/network/routing/vrrp
インタフェースの安定性	非公開

関連項目

vrrpadm(1M), attributes(5)

名前

vscanadm – vscan サービス構成ユーティリティー

形式

```
vscanadm set -p property=value [-p property=value]...
vscanadm get [-p property]...
vscanadm import -p property filename
vscanadm export -p property filename
vscanadm validate -p property filename
vscanadm add-engine [-p property=value]... engine_id
vscanadm remove-engine engine_id
vscanadm set-engine -pproperty=value [-p property=value]... engine_id
vscanadm get-engine [-p property=value]... [engine_id]
vscanadm show
vscanadm stats [-z]
```

機能説明

vscanadm コマンドは、vscan サービス zpool(1M) のプロパティーの設定や表示を行うほか、スキャンの統計情報を提供します。

zfs(1M) などの適切なファイルシステム管理コマンドを使用すると、あるファイルシステムをウィルススキャンから除外するような設定をファイルシステム単位で行うことができます。

スキャンエンジンは外部ホスト上の他社製アプリケーションであり、ファイルに対する実際のウィルススキャン処理を実行します。vscan サービスで使用する複数のスキャンエンジンを構成できます。最低でも 2 つのスキャンエンジンを使用することをお勧めします。ファイルスキャン要求は、負荷分散を実現するために複数の構成済みスキャンエンジンに分配されます。スキャンエンジンはその *engine_id* で識別されます。*engine_id* は、最大 64 バイトのユーザー定義文字列です。

vscan サービスのプロパティーは 2 つのカテゴリに分けられます。1 つはスキャンエンジン定義に固有のスキャンエンジンプロパティー、もう 1 つはサービスに適用され、スキャンエンジン固有ではない一般プロパティーです。

サブコマンド

vscanadm で認証されるサブコマンドは、次のとおりです。

vscanadm set -p *property=value* [-p *property=value*]...
: vscan サービスの一般プロパティーの値を設定します。

  -p *property=value*　プロパティー値を指定します

vscanadm get [-p *property*]...
: vscan サービスの一般プロパティーの値を表示します。プロパティーが指定されなかった場合、vscan サービスのすべての一般プロパティーが表示されます。

  -p *property*　プロパティー値を指定します

`vscanadm set` および `vscanadm get` サブコマンドで使用可能なプロパティーは、次のとおりです。

`max-size`
: ウィルススキャンを実行すべきファイルの最大サイズ。*max-size* を超えるファイルではスキャンが実行されません。*max-size* を超えるファイルへのアクセスを許可するかどうかは、*max-size-action* プロパティーで決まります。

  max-size は数値 (10 進) コンポーネントと単位サイズを表す省略可能な文字コンポーネントから成る文字列で、その形式は「N[.N][KMGTP][B]」です。

  数値コンポーネントのあとには、省略可能な単位を 1 文字、2 文字のいずれかで指定できます。たとえば、K バイトを指定するには「K」、「KB」のいずれかを使用できます。単位指定子は、大文字と小文字は区別されませんが、数値の直後にスペースを入れずに続ける必要があります。

  単位指定子が指定されなかった場合や単位指定子「B」が指定された場合、数値の単位はバイトとみなされます。デフォルト値は 1G バイトです。

  スキャンの最大ファイルサイズは `vscan` サービスで定義されますが、通常はスキャンエンジンでも独自の最大ファイルサイズ設定が定義されます。スキャンエンジンの最大ファイルサイズ以下の値に *max-size* を設定することをお勧めします。

`max-size-action`
: *max-size* より大きいファイルへのアクセスを許可するかどうかを指定します。*max-size* より大きいファイルではウィルススキャンが実行されません。有効な値は次のとおりです。

  allow
  : *max-size* より大きいファイル (ウィルススキャンなし) へのアクセスを許可します。これがデフォルト値です。

  deny
  : *max-size* より大きいファイル (ウィルススキャンなし) へのアクセスを拒否します

`vscanadm import -p` *property filename*
: 指定されたファイルからプロパティー値をインポートします。このファイルには、単一のプロパティー値を指定する行が 1 行だけ含まれている必要があります。

`vscanadm export -p` *property filename*
: 指定されたファイルにプロパティー値をエクスポートします。このファイルには、単一のプロパティー値を指定する行が 1 行だけ含まれている必要があります。

`vscanadm validate -p` *property filename*
指定されたファイル内のプロパティー値を検証します。このファイルには、単一のプロパティー値を指定する行が 1 行だけ含まれている必要があります。

`vscanadm import`、`vscanadm export`、および `vscanadm validate` サブコマンドで使用可能なプロパティーは、次のとおりです。

`types`
ファイルタイプ拡張子マッチングルールのコンマ区切りリスト。このリストは、ウィルススキャンの実行時にどのタイプのファイルをスキャンし、どのファイルを除外すべきかを定義します。各ルールは、ルールインジケータ [+|-] および、ファイルのタイプ拡張子と比較するファイルタイプ *expresssion* から構成されます。ファイルタイプ *expresssion* では大文字と小文字は区別されず、「*」および「?」のワイルドカードを含めることができます。ルールインジケータとファイルタイプ *expresssion* との間にスペースが入らないようにしてください。ファイルタイプ式にコンマが含まれている場合には、そのコンマを「\」(バックスラッシュ) でエスケープする必要があります。ファイルタイプ式には直前のドットは含めません。

ルールインジケータは単一の文字であり、次のいずれかになります。

```
+ include file type in virus scanning
- exclude file type from virus scanning
```

あるファイルがウィルススキャンの候補として評価される際には、そのファイルタイプが types 内に定義されたルールと比較されます。最初に一致したルールが適用されます。一致するルールが 1 つも見つからない場合、そのファイルに対してウィルススキャンが実行されます。types 文字列の合計長が 4096 バイトを超えてはいけません。types リストのデフォルトの内容は、「+*」です。

`vscanadm add-engine [-p` *property=value*`]...` *engine_id*
engine_id で識別される新しいスキャンエンジンを追加します。指定されなかったスキャンエンジンプロパティーでは常に、デフォルト値が使用されます。ホスト名のデフォルト値は、*engine_id* になります。

`-p` *property=value*　　プロパティー値を指定します

`vscanadm remove-engine` *engine_id*
engine_id で識別されるスキャンエンジンとそのすべての構成プロパティー値を削除します。

`vscanadm set-engine -p`*property=value* `[-p` *property=value*`]...` *engine_id*
engine_id で識別されるスキャンエンジンの構成プロパティー値を作成または更新します。

`-p` *property=value*　　プロパティー値を指定します

`vscanadm get-engine` [`-p` *property=value*]... [*engine_id*]

engine_id で識別されるスキャンエンジンの指定されたスキャンエンジンプロパティーの値を表示します。*engine_id* が指定されなかった場合、このサブコマンドは、すべての構成済みスキャンエンジンの指定されたスキャンエンジンプロパティーの値を表示します。プロパティーが指定されなかった場合、このサブコマンドは、`vscan` サービスのすべてのスキャンエンジンプロパティーを表示します。

`-p` *property=value* プロパティー値を指定します

`vscanadm add-engine`、`vscanadm remove-engine`、`vscanadm set-engine`、および `vscanadm get-engine` サブコマンドで使用可能なプロパティーは、次のとおりです。

`enable` スキャンエンジンの有効/無効を指定します。有効な値は「on」(有効) と「off」(無効) です。デフォルトは「on」(有効) です。host プロパティーが無効の場合、スキャンエンジンは有効化できません。

`host` スキャンエンジンのホスト名または IPv4 形式の IP アドレス。

`port` スキャンエンジンの ICAP ポート番号。この数値の範囲は 0 から 65535 です。デフォルトの ICAP ポートは 1344 です。

`max-connection` スキャンエンジンで確立可能な最大同時接続数。この数値の範囲は 1 から 512 までです。このプロパティーのデフォルトは 8 です。

`vscanadm show` `vscan` サービスのすべての一般プロパティーとスキャンエンジンプロパティーの値を表示します。

`vscanadm stats [-z]` `vscan` サービスの次の統計情報を表示またはリセットします。

- スキャンされたファイルの数
- 感染していたファイルの数
- 失敗したスキャン要求の数
- スキャンエラー数 (スキャンエンジンごとのエラー数も含む)

`-z` `vscan` サービスの統計カウンタをゼロにリセットします

使用例

例 1 最大サイズ制限の設定

ウィルススキャンの対象となるファイルの最大サイズ制限を 128M バイトに設定するには、次のように入力します。

```
# vscanadm set -p max-size=128M
```

例 2 ファイルへのアクセス許可

最大ファイルサイズを超えるファイルへのアクセスを許可するには、次のように入力します。

```
# vscanadm set -p max-size-action=allow
```

例 3 ファイルタイプの設定

タイプ「odt」、「exe」、「jpg」のファイルのみがウィルススキャンの対象となるように types を設定するには、次のように入力します。

```
# vscanadm set -p types=+odt,+exe,+jpg,-*
```

「doc」以外のすべてのファイルタイプがウィルススキャンの対象となるように types を設定するには、次のように入力します。

```
# vscanadm set -p types=-doc,+*
```

例 4 ファイルタイプリストの表示

ファイルタイプのリストを表示するには、次のように入力します。

```
# vscanadm get -p types
```

例 5 スキャンエンジンの追加

スキャンエンジン「my_eng」をデフォルト値を使用して追加するには、次のように入力します。

```
# vscanadm add-engine my_eng
```

例 6 スキャンエンジンの無効化

スキャンエンジン「my_eng」を無効にするには、次のように入力します。

```
# vscanadm set-engine -p enable=off my_eng
```

例 7 スキャンエンジンのプロパティーの表示

スキャンエンジン「my_eng」のプロパティーを表示するには、次のように入力します。

```
# vscanadm get-engine my_eng
```

例 8 スキャンエンジンの削除

スキャンエンジン「my_eng」を削除するには、次のように入力します。

```
# vscanadm remove-engine my_eng
```

例 9　vscan サービスの一般プロパティーとスキャンエンジンプロパティーの表示

vscan サービスのすべての一般プロパティーとスキャンエンジンプロパティーを表示するには、次のように入力します。

```
# vscanadm show
```

終了ステータス

次の終了値が返されます。

0　　正常終了。

0 以外　　エラーが発生しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/virus-scan
インタフェースの安定性	不確実
ユーティリティーの出力形式	インタフェースではない

関連項目

vscand(1M), zfs(1M), attributes(5), smf(5)

注意事項

vscanadm を使用して vscan プロパティーや統計情報を表示する権限は、すべてのユーザーに対して許可されます。プロパティー値を設定したり統計情報をリセットしたりするには、次の承認が必要となります。

solaris.smf.value.vscan
　プロパティー値を変更、または統計情報をリセットします

solaris.manage.vscan
　サービスを更新してプロパティー値の変更内容を反映します

プロパティーの追加や削除 (add-engine、 remove-engine) を行うには、次の承認が必要となります。

solaris.smf.modify.application　　プロパティーグループを追加または削除します

solaris.manage.vscan　　サービスを更新してプロパティー値の変更内容を反映します

「VSCAN Management」プロファイルには、これらの承認がすべて含まれています。

名前

vscand – vscan サービスデーモン

形式

`/usr/lib/vscan/vscand`

機能説明

vscand は、ファイルのオープン/クローズ処理時にファイルシステムからのウィルススキャン要求を処理するデーモンです。ファイルシステムによっては、そのファイルシステムの管理コマンド (zfs(1M) など) を使用してウィルススキャンの有効化や無効化をデータセット単位で行える場合があります。

ファイル状態またはスキャンポリシー (vscanadm(1M) を参照) からファイルのスキャンが要求された場合、vscand は ICAP (Internet Content Adaptation Protocol、RFC 3507) を使用して他社製の外部ウィルススキャナ (スキャンエンジン) と通信し、そのファイルをスキャンします。

ファイルがスキャンエンジンに送られるのは、そのファイルが前回のスキャン後に変更された場合と、そのファイルがまだ最新のスキャンエンジン構成 (ウィルス定義) に基づいてスキャンされていない場合です。この情報の格納先として、ファイルの modified 属性と scanstamp 属性が使用されます。ファイルのスキャンが完了すると、modified 属性はクリアされ、scanstamp 属性は更新されます。

ファイルにウィルスが含まれることが判明した場合、ウィルスの情報が syslogd(1M) に記録され、監査レコードが書き込まれ、そのファイルは (quarantine 属性の設定によって) 隔離されます。ファイルが隔離されると、そのファイルの読み取り、実行、または名前変更の試みがファイルシステムによって拒否されます。syslogd(1M) エントリと監査レコードには、感染したファイルの名前とそのファイル内で検出された違反の情報が記録されています。各違反は「ID - 脅威の説明」として指定されますが、この ID と脅威の説明については、ICAP RFC 3507 の拡張 X-Infection-Found-Header 内で定義されています。

デフォルトでは、vscand はポート 1344 上でスキャンエンジンに接続します。ポートなどのサービス構成パラメータを構成するには、vscanadm(1M) を使用します。

vscan サービスは、デフォルトでは無効になっていますが、svcadm(1M) を使用すると有効化できます。

終了ステータス

次の終了値が返されます。

0	デーモンが正常に起動されました。
0 以外	デーモンの起動に失敗しました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/storage/virus-scan
インタフェースの安定性	不確実

関連項目

ps(1), svcs(1), logadm(1M), svcadm(1M), syslogd(1M), vscanadm(1M), zfs(1M), attributes(5), smf(5)

注意事項

ファイルへのアクセス時に、NFSv3 のような、ファイルシステムのオープン/クローズ処理を呼び出さないプロトコルが使用された場合、そのファイルではウィルススキャンは起動されません。

ファイルの内容は「平文」データとしてスキャンエンジンに転送されます。

有効化、無効化、または再起動要求など、vscan サービスに関する管理操作は、svcadm(1M) を使用して実行できます。vscan サービスのステータスを照会するには、svcs(1) コマンドを使用します。

vscan サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/filesystem/vscan
```

名前

vtdaemon – テキスト仮想コンソール間の切り替え機能

形式

/usr/lib/vtdaemon

機能説明

vtdaemon は、複数のテキスト仮想コンソール間のセキュリティー保護された切り替え機能をホットキー経由で提供します。これは完全な特権を使用して実行されます。

このサービスは、Device Security 権利プロファイルに含まれるアクション承認 solaris.smf.manage.vt を使用して管理されます。このサービスの対象はローカルのみであり、インバウンドネットワークポートは用意されていません。このサービスは、非大域ゾーン内で無効化されるほか、仮想コンソール機能が使用できない場合にも無効化されます。

vtdaemon では認証時に PAM が使用されます。

vtdaemon はテキストコンソールのロックやロック解除を行うだけです。グラフィカルな X セッションの場合、認証や監査を処理する対応するロックプログラム (xscreensaver など) が呼び出されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/core-os
インタフェースの安定性	非公開

関連項目

svcs(1), svcadm(1M), attributes(5), smf(5)

注意事項

vtdaemon サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

svc:/system/vtdaemon:default

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

vtdaemon サービスの SMF 構成プロパティーは、次のとおりです。

hotkeys (ブール (論理) 型)
: 承認ユーザーがホットキー経由の VT 切り替えを動的に有効化または無効化できるようにします。そのデフォルト値は TRUE (有効) です。

secure (ブール型)
: 承認ユーザーがホットキーのセキュリティーを動的に有効化または無効化できるようにします。無効化されると、ユーザーは任意のセッションへの切り替えを認証なしに自由に行えるようになります。そのデフォルト値は TRUE (有効) です。

名前

wall – すべてのユーザーへの通知

形式

`/usr/sbin/wall [-a] [-g grpname] [filename]`

機能説明

`wall` はファイルの終わりまで標準入力を読み取ります。その後、現在ログインしているすべてのユーザーに以下の文字で始まるメッセージを送信します。

```
Broadcast Message from . . .
```

filename が指定されていれば、メッセージはこのファイルから読み込まれます。通常、リモートログインセッションに対応していない仮想端末は無視されます。したがって、ウィンドウシステムを使用しているときは、メッセージはコンソールウィンドウにだけに表示されます。ただし、`-a` オプションを指定すると、メッセージを仮想端末などにも送信します。

一般的には、システムをシャットダウンする前に、すべてのユーザーに警告を与える場合に、このメッセージを用います。

ユーザーが起動させた保護を無効にする場合、送信側はスーパーユーザーである必要があります (`mesg(1)` 参照)。

`wall` は、他のユーザーの端末上で書き込み権を持つために、グループ ID `tty` に対して `setgid()` を実行します。 (`setuid(2)` 参照)。

`wall` は、ユーザーの端末に送信する前に非表示可能文字を検出します。制御文字は、適切な ASCII 文字が後に続く ^ として表示されます。すなわち、高位ビットが設定された文字は meta 表記法で表示されます。たとえば、'`\003`' は '`^C`' と表示され、'`\372`' は '`M-z`' と表示されます。

オプション

次のオプションがサポートされています。

`-a`	コンソールと仮想端末にメッセージを送ります。
`-g` *grpname*	グループデータベース (`group(4)` を参照) ごとに、grpname で指定されたグループのユーザーだけにメッセージを送ります。

環境変数

`LC_*` 変数 (`LC_CTYPE`、`LC_TIME`、`LC_COLLATE`、`LC_NUMERIC`、`LC_MONETARY`) が環境に設定されていなければ、それぞれ対応する ロケール のカテゴリにおける `wall` の動作は、環境変数 `LANG` によって決定されます。 (`environ(5)` 参照)。もし、`LC_ALL` が設定されていれば、その内容が `LANG` 変数やその他の `LC_*` 変数より優先されます。前述の変数が環境にまったく設定されていなければ、 C ロケール (米国スタイル) が `wall` の動作を決定します。

ファイル

`/dev/tty*`

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値

使用条件	system/core-os

関連項目

mesg(1), write(1), setuid(2), attributes(5), environ(5)

注意事項

wall は、ユーザーの tty ファイルでオープンが失敗すると、“Cannot send to ..” と表示します。

名前

wanboot_keygen – create and display client and server keys for WAN booting

形式

```
/usr/lib/inet/wanboot/keygen -c -o net=a.b.c.d ,cid=client_ID,type=3des
/usr/lib/inet/wanboot/keygen -c -o net=a.b.c.d ,cid=client_ID,type=aes
/usr/lib/inet/wanboot/keygen -m
/usr/lib/inet/wanboot/keygen -c -o net=a.b.c.d ,cid=client_ID,type=sha1
/usr/lib/inet/wanboot/keygen -d -m
/usr/lib/inet/wanboot/keygen -c -o net=a.b.c.d ,cid=client_ID,type=keytype
```

機能説明

The `keygen` utility has three purposes:

- Using the `-c` flag, to generate and store per-client 3DES/AES encryption keys, avoiding any DES weak keys.
- Using the `-m` flag, to generate and store a “master” HMAC SHA-1 key for WAN install, and to derive from the master key per-client HMAC SHA-1 hashing keys, in a manner described in RFC 3118, Appendix A.
- Using the `-d` flag along with either the `-c` or `-m` flag to indicate the key repository, to display a key of type specified by *keytype*, which must be one of `3des`, `aes`, or `sha1`.

The `net` and `cid` arguments are used to identify a specific client. Both arguments are optional. If the `cid` option is not provided, the key being created or displayed will have a per-network scope. If the `net` option is not provided, then the key will have a global scope. Default net and code values are used to derive an HMAC SHA-1 key if the values are not provided by the user.

オプション

The following options are supported:

`-c` Generate and store per-client 3DES/AES encryption keys, avoiding any DES weak keys. Also generates and stores per-client HMAC SHA-1 keys. Used in conjunction with `-o`.

`-d` Display a key of type specified by *keytype*, which must be one of `3des`, `aes`, or `sha1`. Use `-d` with `-m` or with `-c` and `-o`.

`-m` Generate and store a “master” HMAC SHA-1 key for WAN install.

`-o` Specifies the WANboot client and/or keytype.

使用例

例 1 Generate a Master HMAC SHA-1 Key

```
# keygen -m
```

例 2 Generate and Then Display a Client-Specific Master HMAC SHA-1 Key

```
# keygen -c -o net=172.16.174.0,cid=010003BA0E6A36,type=sha1
# keygen -d -c -o net=172.16.174.0,cid=010003BA0E6A36,type=sha1
```

例 3 Generate and Display a 3DES Key with a Per-Network Scope

```
# keygen -c -o net=172.16.174.0,type=3des
# keygen -d -o net=172.16.174.0,type=3des
```

終了ステータス

`0` Successful operation.

`>0` An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Obsolete

関連項目

`attributes(5)`

名前

wanboot_keymgmt – insert and extract keys

形式

```
/usr/lib/inet/wanboot/keymgmt -i -k key_file -s keystore -o type=keytype
/usr/lib/inet/wanboot/keymgmt -x -f outfile -s keystore -o type=keytype
```

機能説明

The `keymgmt` utility has two purposes:

- To take a raw key, stored in *key_file*, and insert it in the repository specified by *keystore*.
- To extract a key of a specified type from the repository specified by *keystore*, depositing it in *outfile*.

outfile will be created if it does not already exist. The type of key being added or extracted is specified by *keytype* and may have one of four values: `3des`, `aes`, `rsa`, or `sha1` (the last used by HMAC SHA-1). When extracting a key, the first key with an OID matching the supplied type is used.

Arguments

The following arguments are supported:

`-i`	Used in conjunction with `-k` to insert a raw key in *keystore*.
`-f` *outfile*	Used to specify a file to receive an extracted key.
`-k` *key_file*	Used in conjunction with `-i` to specify the file in which a raw key is stored. This key will be inserted in *keystore*.
`-o` type=*keytype*	Specifies the type of key being inserted or extracted. Must be one of `3des`, `aes`, `rsa`, or `sha1`.
`-s` *keystore*	Specifies a repository in which a key will be inserted or from which a key will be extracted.
`-x`	Used in conjunction with `-f` to extract a key of a specified type and deposit it in *outfile*.

終了ステータス

`0`	Successful operation.
`>0`	An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Obsolete

関連項目

`attributes(5)`

ITU-T Recommendation X.208

名前

wanboot_p12split – split a PKCS #12 file into separate certificate and key files

形式

```
/usr/lib/inet/wanboot/p12split -i p12file -c out_cert -k out_key
     [-t out_trust -l id -v]
```

機能説明

The `p12split` utility extracts a certificate and private key from the repository specified by *p12file*, depositing the certificate in *out_cert* and the key in *out_key*. If supplied, the `-l` option specifies the value for the `LocalKeyId` that will be used in the new certificate and key files. `p12split` can optionally extract a trust certificate into the *out_trust* file if the `-t` option is specified. Use the `-v` option to get a verbose description of the split displayed to standard output.

オプション

The following arguments and options are supported:

`-c` *out_cert*	Specifies a repository that receives a extracted certificate.
`-i` *p12file*	Specifies a repository from which a certificate and private key is extracted.
`-k` *out_key*	Specifies a repository that receives a extracted private key.
`-l` *id*	Specifies the value for the `LocalKeyId` that will be used in the new certificate and key files.
`-t` *out_trust*	Specifies a file for receiving an extracted trust certificate.
`-v`	Displays a verbose description of the split to stdout.

終了ステータス

`0`	Successful operation.
`>0`	An error occurred.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Uncommitted

関連項目

`attributes(5)`

名前

wanbootutil – manage keys and certificates for WAN booting

形式

`wanbootutil [keygen] [`*option_specific_arguments*`]`

`wanbootutil [keymgmt] [`*option_specific_arguments*`]`

`wanbootutil [p12split] [`*option_specific_arguments*`]`

機能説明

The `wanbootutil` command creates and manages WANboot encyrption and hashing keys and manipulates PKCS #12 files for use by WAN boot.

`wanbootutil` has three subcommands, each covered in a separate man page:

`wanboot_keygen(1M)`	Generates encryption and hashing keys.
`wanboot_keymgmt(1M)`	Inserts and extracts keys from WAN boot key repositories.
`wanboot_p12split(1M)`	Splits a PKCS #12 file into separate certificate and key files for use by WAN boot.

オプション

The options are supported for `wanbootutil` are the use of `keygen`, `keymgmt`, or `p12split`. The options for these subcommands are described in their respective man pages.

使用例

例 1 Generate a 3DES Client Key

```
# wanbootutil keygen -c -o net=172.16.174.0,cid=010003BA0E6A36,type=3des
```

例 2 Insert an RSA Private Client Key

```
wanbootutil keymgmt -i -k keyfile \
  -s /etc/netboot/172.16.174.0/010003BA0E6A36/keystore -o type=rsa
```

例 3 Split a PKCS #12 File into Certificate and Key Components

```
# wanbootutil p12split -i p12file -c out_cert -k out_key
```

終了ステータス

`0`	Successful operation.
`non-zero`	An error occurred. Writes an appropriate error message to standard error.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/boot/wanboot
Interface Stability	Obsolete

関連項目

`wanboot_keygen(1M)`, `wanboot_keymgmt(1M)`, `wanboot_p12split(1M)`, `attributes(5)`

名前

wbemadmin – start Sun WBEM User Manager

形式

`/usr/sadm/bin/wbemadmin`

機能説明

The `wbemadmin` utility starts Sun WBEM User Manager, a graphical user interface that enables you to add and delete authorized WBEM users and to set their access privileges. Use this application to manage access to groups of managed resources, such as disks and installed software, in the Solaris operating environment.

The `wbemadmin` utility allows you to perform the following tasks:

Manage user access rights	Use the `wbemadmin` utility to add, delete, or modify an individual user's access rights to a namespace on a WBEM-enabled system.
Manage namespace access rights	Use the `wbemadmin` utility to add, delete, or modify access rights for all users to a namespace.

The Sun WBEM User Manager displays a Login dialog box. You must log in as root or a user with write access to the `root\security` namespace to grant access rights to users. By default, Solaris users have guest privileges, which grants them read access to the default namespaces.

Managed resources are described using a standard information model called Common Information Model (CIM). A CIM object is a computer representation, or model, of a managed resource, such as a printer, disk drive, or CPU. CIM objects can be shared by any WBEM-enabled system, device, or application. CIM objects are grouped into meaningful collections called schema. One or more schemas can be stored in directory-like structures called namespaces.

All programming operations are performed within a namespace. Two namespaces are created by default during installation:

- `root\cimv2` — Contains the default CIM classes that represent objects on your system.
- `root\security` — Contains the security classes used by the CIM Object Manager to represent access rights for users and namespaces.

When a WBEM client application connects to the CIM Object Manager in a particular namespace, all subsequent operations occur within that namespace. When you connect to a namespace, you can access the classes and instances in that namespace (if they exist) and in any namespaces contained in that namespace.

When a WBEM client application accesses CIM data, the WBEM system validates the user's login information on the current host. By default, a validated WBEM user is granted read access to the Common Information Model (CIM) Schema. The CIM Schema describes managed objects on your system in a standard format that all WBEM-enabled systems and applications can interpret.

You can set access privileges on individual namespaces or for a user-namespace combination. When you add a user and select a namespace, by default the user is granted read access to CIM objects in the selected namespace. An effective way to combine user and namespace access rights is to first restrict access to a namespace. Then grant individual users read, read and write, or write access to that namespace.

You cannot set access rights on individual managed objects. However you can set access rights for all managed objects in a namespace as well as on a per-user basis.

If you log in to the root account, you can set the following types of access to CIM objects:

- Read Only — Allows read-only access to CIM Schema objects. Users with this privilege can retrieve instances and classes, but cannot create, delete, or modify CIM objects.
- Read/Write — Allows full read, write, and delete access to all CIM classes and instances.
- Write — Allows write and delete, but not read access to all CIM classes and instances.
- None — Allows no access to CIM classes and instances.

Context help is displayed in the left side of the `wbemadmin` dialog boxes. When you click on a field, the help content changes to describe the selected field. No context help is available on the main User Manager window.

The `wbemadmin` security administration tool updates the following Java classes in the `root\security` namespace:

- `Solaris_UserAcl` — Updated when access rights are granted or changed for a user.
- `Solaris_namespaceAcl` — Updated when access rights are granted or changed for a namespace.

使用法

The `wbemadmin` utility is not the tool for a distributed environment. It is used for local administration on the machine on which the CIM Object Manager is running.

終了ステータス

The `wbemadmin` utility terminates with exit status `0`.

Warning

The `root\security` namespace stores access privileges. If you grant other users access to the `root\security` namespace, those users can grant themselves or other users rights to all other namespaces.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目 mofcomp(1M), wbemlogviewer(1M), init.wbem(1M), attributes(5)

名前

wbemconfig – convert a JavaSpaces datastore to the newer Reliable Log datastore format

形式

```
/usr/sadm/lib/wbem/wbemconfig convert
```

機能説明

A Reliable Log directory is created that contains the converted data. This directory is named `/var/sadm/wbem/logr`.

The `convert` argument is the only supported option of this command. You should only run this command after stopping WBEM (CIM Object Manager) with the `init.wbem stop` command. Otherwise your data may be corrupted.

This command successfully converts any proprietary custom MOFs you have created in the datastore, but not any CIM or Solaris MOFs you have modified. These will be destroyed. To recompile any modified CIM or Solaris MOFs into the new datastore, run the `mofcomp` command on the MOF files containing the class definitions.

Because the `wbemconfig convert` command invokes the JVM (Java Virtual Machine) to perform conversion of the JavaSpaces datastore, you must be running the same version of the JVM as when the original JavaSpaces storage was created. After the `wbemconfig convert` command is completed, you can change to any version of the JVM you want.

To see what version of the JVM you are running, issue the `java -version` command.

オプション

The following options are supported:

convert　Convert a JavaSpaces datastore to the newer Reliable Log datastore format.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

`init.wbem(1M)`, `wbemadmin(1M)`, `wbemlogviewer(1M)`, `mofcomp(1M)`, `attributes(5)`

名前

wbemlogviewer – start WBEM Log Viewer

形式

`/usr/sadm/bin/wbemlogviewer`

機能説明

The `wbemlogviewer` utility starts the WBEM Log Viewer graphical user interface, which enables administrators to view and maintain log records created by WBEM clients and providers. The WBEM Log Viewer displays a Login dialog box. You must log in as root or a user with write access to the `root\cimv2` namespace to view and maintain log files. Namespaces are described in `wbemadmin(1M)`.

Log events can have three severity levels.

- Errors
- Warnings
- Informational

The WBEM log file is created in the `/var/sadm/wbem/log` directory, with the name `wbem_log`. The first time the log file is backed up, it is renamed `wbem_log.1`, and a new `wbem_log` file is created. Each succeeding time the `wbem_log` file is backed up, the file extension number of each backup log file is increased by 1, and the oldest backup log file is removed *if* the limit, which in turn is specified in the log service settings, on the number of logfiles is exceeded. Older backup files have higher file extension numbers than more recent backup files.

The log file is renamed with a `.1` file extension and saved when one of the following two conditions occur:

- The current file reaches the specified file size limit.
- A WBEM client application uses the `clearLog()` method in the `Solaris_LogService` class to clear the current log file.
- A WBEM client application uses the `clearLog()` method in the Solaris_LogService class to clear the current log file.
- A user chooses Action->Back Up Now in the Log Viewer application.

Help is displayed in the left panel of each dialog box. Context help is not displayed in the main Log Viewer window.

使用法

The WBEM Log Viewer is not the tool for a distributed environment. It is used for local administration.

The WBEM Log Viewer allows you to perform the following tasks:

View the logs

Set properties of log files	Click Action->Log File Settings to specify log file parameters and the log file directory.
Back up a log file	Click Action->Back Up Now to back up and close the current log file and start a new log file.

Open historical log files	Click Action->Open Log File to open a backed-up log file.
Delete an old log file	Open the file and then click Action->Delete Log File. You can only delete backed-up log files.
View log record details	Double-click a log entry or click View->Log Entry Details to display the details of a log record.
Sort the logs	Click View->Sort By to sort displayed entries. You can also click any column heading to sort the list. By default, the log entries are displayed in reverse chronological order (new logs first).

終了ステータス

The `wbemlogviewer` utility terminates with exit status `0`.

ファイル

`/var/sadm/wbem/log/wbem_log` WBEM log file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWwbco

関連項目

`wbemadmin(1M)`, `init.wbem(1M)`, `mofcomp(1M)`, `attributes(5)`

名前

wcadmin – manage the configuration of the Sun Java Web Console

形式

`/usr/sbin/wcadmin` *subcommand options*

`/usr/sbin/wcadmin` *[-h| --help]*

`/usr/sbin/wcadmin` *[-V | --version]*

機能説明

`wcadmin` is a Command Line Interface (CLI) — based tool for managing the configuration of the Sun Java Web Console.

Subcommands

The following subcommands are supported:

add

The `wcadmin add` subcommand adds a new shared jar file, a new JAAS login module, or a new shared service property to the console configuration. An optional identifier may be specified; if omitted, the identifier is derived from the resource name. The resource is added when the console is next started.

The format of the `add` subcommand is:

add `-l -a` *appname* [`-n` *id*] *jarpath*

add `-m -a` *appname* [`-n` *id*] `-b` *behavior* `-s` *service* [`-o` *name=value*] *classname*

add `-p -a` *appname name=value* [... *name=value*]

deploy

The `wcadmin deploy` subcommand deploys the specified web application into the console's web server instance. Applications are deployed directly from their installation directories.

The format of the `deploy` subcommand is:

deploy [`-D`] `-a` *appname* `-x` *context app_path*

disable

The `wcadmin disable` subcommand disables access to the specified web application in the console's web server instance.

The format of the `disable` subcommand is:

disable `-x` *context*

enable

The `wcadmin enable` subcommand enables access to the specified web application in the console's web server instance.

The format of the `enable` subcommand is:

enable `-x` *context*

list

The `wcadmin list` subcommand lists the resources currently configured for the console; including deployed web applications, shared jar files, login modules, and shared service properties. If no option is specified, all resources are listed.

The format of the `list` subcommand is:

list [`-a`] [`-d`] [`-l`] [`-m`] [`-p`]

password — The `wcadmin password` subcommand manages the administrator and security keystore passwords for the console. Keystore passwords should not be changed while the console is running.

The format of the `password` subcommand is:

password [`-a`] [`-k`] [`-t`]

password `-f` *password_file*

reload — The `wcadmin reload` subcommand unloads the specified web application from the console's web server instance and reloads the application from its original installation directory.

The format of the `reload` subcommand is:

reload `-x` *context*

remove — The `wcadmin remove` subcommand removes a shared jar file, a login module, or a shared service property from the console configuration. The resource may be specified by its identifier or by its full jarpath or classname. The resource is removed when the console is next started.

The format of the `remove` subcommand is:

remove `-l -a` *appname* [`-n` *id*] *jarpath*

remove `-m -a` *appname* [`-n` *id*] *classname*

remove `-p -a` *appname property* [...]

undeploy — The `wcadmin undeploy` subcommand undeploys the specified web application from the console's web server instance.

The format of the `undeploy` subcommand is:

undeploy [`-D`] `-a` *appname* `-x` *context*

オプション

The following options are supported:

`-h` | `--help` | `-?`	Display runtime help.
`-V` | `--version`	Display console version information.
`-D` | `--defer`	When used with the deploy and undeploy subcommands, defers the deployment or undeployment until the next console restart.

The operation is deferred by simply adding or removing the corresponding resource registration notification file. If defer is not specified, a runtime deployment or undeployment is performed, so that the application becomes available or unavailable in the running console. If the console instance is not currently running, the operation is automatically deferred.

`-a|--adminpassword` Specify that the administrator password should be changed, when used with password subcommand. You are prompted for a new password, which must be 8 to 32 characters.

`-a|--application` Specify the application name, when used with subcommands other than password subcommand.

`-d|--detail` Specify that configuration details of each resource should be displayed.

`-f` Specify the fully qualified path name to a file containing one or more password property values. See the description of the password_file argument.

`-k|--keypassword` Specify that the keystore password should be changed. You are prompted for a new password, which must be 8 to 32 characters.

`-l|--library` Specify that the resource is a shared jar file.

`-m|--module` Specify that the resource is a JAAS login module.

`-n|--name` Specify the short-hand identifier name for the resource. If omitted, the identifier name is derived from the full resource name.

`-o|--option` Specify the name and value of a login module option property, separated by the equals character.

`-p|--property` Specify that the resource is one or more shared service properties.

`-s|--service` Specify the name of the JAAS login service definition. If omitted, the default console login service definition is assumed.

`-t|--trustpassword` Specify that the truststore password should be changed. You are prompted for a new password, which must be 8 to 32 characters.

`-x|--context` Specify the web application context path name under which the application is deployed.

Arguments

app_path The fully qualified file system path to the web application installation directory.

appname The application name. The name must be unique among all web applications registered with the console. It is also used as the name of the subdirectory under the console's pre-registration directory which contains

all the resource registration notification files for that application. Typically, the application package name, plugin identifier, or context path name is specified for the application name.

behavior The JAAS login module control flag behavior. Must be one of “optional”, “required”, “requisite”, or “sufficient”.

classname The fully qualified Java package class name of the JAAS login module. The specified class must be included in a shared jar file added to the console.

context The web application context path name under which the application is deployed. With the .reg suffix, the context forms the file name of the registration notification file for that application.

id The short-hand identifier name for a jar file or login module resource to be added or removed. The identifier name must be unique among the resources shared for a given application name. With the .reg suffix, it forms the file name of the registration notfication file for that resource.

jarpath The fully qualified file system path to the jar file resource. When the resource is added to the console, its path is included in the classpath of the console's shared class loader.

option The JAAS login module option property name.

property The shared service property name.

password_file The fully qualified path to a password text file that contains the new administrator, keystore, and truststore passwords in property file format. The administrator password is specified using the “adminpassword” property. The keystore password is specified using the “keypassword” property. The truststore password is specified using the “trustpasswd” property. At least one password property must be contained in the password file.

value The login module option or shared service property value. If the value contains white space, it must be quoted.

使用例

The following command adds a jar file to be shared in the console:

```
wcadmin add -l -a myapp_1.0 -n wbem /usr/sadm/lib/wbem.jar
```

The following command deploys a new web application:

```
wcadmin deploy -a myapp_1.0 -x myapp /opt/SUNWmyapp/myapp
```

The following command reloads an existing web application:

```
wcadmin reload -x myapp
```

The following command undeploys a web application at the next server restart:

```
wcadmin undeploy -D -a myapp_1.0 -x myapp
```

The following command lists all the deployed web applications in the console. If the status field is “running”, the web application is available. If the status field is “stopped”, the web application is disabled and is not available. If all web applications are “stopped”, this typically indicates the console web server instance is not running.

```
wcadmin list -a
```

The following command removes a shared jar file:

```
wcadmin remove -l -a myapp_1.0 -n wbem
```

The following command changes passwords that are specified in a file:

```
wcadmin password -f /home/mydir/console-passwords
```

終了ステータス

The following exit values are returned:

- `0` Subcommand succeeded without error
- `1` Usage error: missing or malformed arguments
- `2` Fatal error: subcommand failed with one or more errors

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWmcon
Interface Stability	Committed

関連項目

`attributes(5)`, `environ(5)`

名前

whodo – who is doing what

形式

`/usr/sbin/whodo [-h] [-l] [user]`

機能説明

The `whodo` command produces formatted and dated output from information in the `/var/adm/utmpx` and `/proc/pid` files.

The display is headed by the date, time, and machine name. For each user logged in, device name, user-ID and login time is shown, followed by a list of active processes associated with the user-ID. The list includes the device name, process-ID, CPU minutes and seconds used, and process name.

If *user* is specified, output is restricted to all sessions pertaining to that user.

オプション

The following options are supported:

`-h` Suppress the heading.

`-l` Produce a long form of output. The fields displayed are: the user's login name, the name of the tty the user is on, the time of day the user logged in (in *hours*:*minutes*), the idle time — that is, the time since the user last typed anything (in *hours*:*minutes*), the CPU time used by all processes and their children on that terminal (in *minutes*:*seconds*), the CPU time used by the currently active processes (in *minutes*:*seconds*), and the name and arguments of the current process.

使用例

例 1 Using the whodo Command

The command:

```
example% whodo
```

produces a display like this:

```
Tue Mar 12 15:48:03 1985
bailey
tty09    mcn       8:51
    tty09   28158    0:29 sh

tty52    bdr      15:23
    tty52   21688    0:05 sh
    tty52   22788    0:01 whodo
    tty52   22017    0:03 vi
    tty52   22549    0:01 sh

xt162    lee      10:20
    tty08    6748    0:01 layers
    xt162    6751    0:01 sh
    xt163    6761    0:05 sh
    tty08    6536    0:05 sh
```

環境変数

If any of the LC_* variables (LC_CTYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY) (see environ(5)) are not set in the environment, the operational behavior of tar(1) for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents are used to override both the LANG and the other LC_* variables. If none of the above variables is set in the environment, the "C" (U.S. style) locale determines how whodo behaves.

LC_CTYPE
: Determines how whodo handles characters. When LC_CTYPE is set to a valid value, whodo can display and handle text and filenames containing valid characters for that locale. The whodo command can display and handle Extended Unix code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. whodo can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

LC_MESSAGES
: Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

LC_TIME
: Determines how whodo handles date and time formats. In the "C" locale, date and time handling follow the U.S. rules.

終了ステータス

The following exit values are returned:

0 Successful completion.

non-zero An error occurred.

ファイル

/etc/passwd System password file

/var/adm/utmpx User access and administration information

/proc/pid Contains PID

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

ps(1), who(1), attributes(5), environ(5)

名前

wpad – WPA および WPA2 プロトコルデーモン

形式

`/usr/lib/inet/wpad [-i` *interface*`] [-k` *pre_shared_key_name*`]`

機能説明

`wpad` デーモンは、IEEE802.11i 標準で規定されている WiFi Protected Access (WPA) Version 1 および 2 の共通クライアント機能を提供します。WPA は、業界団体の WiFi Alliance によって作成されました。WPA は IEEE 802.11i 標準の大部分を実装しており、802.11i の準備が整うまでの間、WEP (Wired Equivalent Privacy) の代わりに使用するために過渡的な手段として用意されたものです。WPA2 は標準を完全に実装したものです。

`wpad` が提供する WPA/IEEE 802.11i 機能は、次のとおりです。

- WPA-PSK (「WPA-Personal」)
- CCMP、TKIP、WEP104、WEP40 の鍵管理

`wpad` デーモンの停止や開始を行うには、`dladm(1M)` を使用します。次を使用すると、

```
# dladm connect-wifi
```

`wpad` デーモンが開始されます。次を使用すると、

```
# dladm disconnect-wifi
```

デーモンが停止されます。

オプション

次のオプションがサポートされています。

`-i` *interface*
: `wpad` デーモンを開始する WiFi リンクインタフェースを指定します。

`-k` *pre_shared_key_name*
: WiFi リンクで使用される事前共有鍵を指定します。

使用例

例 1　特定の WiFi リンクでの wpad デーモンの開始

WPA 鍵 `psk` を作成するには、次のコマンドを入力します。

```
# dladm create-secobj -c wpa psk
```

鍵 `psk` を使用してリンク `ath0` の ESSID `wlan` に接続するには、次のコマンドを入力します。

```
# dladm connect-wifi -k psk -e wlan ath0
```

例 2　特定の WiFi リンクでの wpad デーモンの停止

リンク `ath0` のデーモンを停止するには、次のように入力します。

```
# dladm disconnect-wifi ath0
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	service/network/wpa
インタフェースの安定性	不確実

関連項目

svcs(1), dladm(1M), svcadm(1M), attributes(5), smf(5)

注意事項

wpad サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

svc:/network/wpa:<*link*>

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcadm(1M) コマンドを使用します。

名前

wracct – write extended accounting records for active processes and tasks

形式

```
/usr/sbin/wracct -i id_list [-t record_type]
     {process | task}
```

機能説明

The `wracct` utility allows the administrator to invoke the extended accounting system, if active, to write intermediate records representing the resource usage of a selected set of processes or tasks. For tasks, a *record_type* option is also supported, allowing the administrator to request the writing of:

- an interval record, which reflects task usage since a previous interval record (or since task creation if there is no interval record), or
- a partial record, which reflects usage since task creation.

オプション

The following options are supported:

`-i` *id_list*
: Select the IDs of the tasks or processes to write records for. Specify *id_list* as a comma- or space-separated list of IDs, presented as a single argument. For some shells, this requires appropriate quoting of the argument.

`-t` *record_type*
: Select type of record to write for the selected task or process. For tasks, *record_type* can be `partial` or `interval`. `partial` is the default type, and the only type available for process records.

オペランド

The following operands are supported:

`process`
: Treat the given ID as a process ID for the purposes of constructing and writing an extended accounting record.

`task`
: Treat the given ID as a task ID for the purposes of constructing and writing an extended accounting record.

使用例

例 1 Writing a Partial Record

Write a partial record for all active `sendmail` processes.

```
# /usr/sbin/wracct -i "`pgrep sendmail`" process
```

例 2 Writing an Interval Record

Write an interval record for the task with `ID 182`.

```
# /usr/sbin/wracct -t interval -i 182 task
```

終了ステータス

The following exit values are returned:

`0`
: Successful completion.

`1`
: An error occurred.

`2`
: Invalid command line options were specified.

3 Pertinent components of extended accounting facility are not active.

ファイル

`/var/adm/exacct/task`

`/var/adm/exacct/proc` Extended accounting data files.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

関連項目

`acctadm(1M)`, `attributes(5)`

名前

wusbadm – ワイヤレス USB ホストおよびデバイスの管理

形式

```
wusbadm list [-h | -d] [-o field[,...]]

wusbadm associate [-h host-id] [[-c [-f]] | -n] [-o]

wusbadm remove-dev [[-d dev-id] | [-h host-id]] [-f]

wusbadm remove-host [-h host-id] [-f]

wusbadm enable-host [-h host-id]

wusbadm disable-host [-h host-id]
```

機能説明

wusbadm コマンドは、ホストとデバイスの情報の一覧表示、ホストとデバイスの関連付け、ホストまたはデバイスの情報のシステムからの削除、ホストの有効化または無効化など、ワイヤレス USB ホストおよびデバイスの管理作業を行うためのコマンド行インタフェースを提供します。

ワイヤレス USB デバイスをホストにはじめて接続する前に、wusbadm の associate サブコマンドを実行してそれらの間の関連付け情報を設定する必要があります。その後、デバイスの無線を単純にオン/オフすることでデバイスの接続/切断を行えます。メーカーによって、デバイスにボタンが付いている場合があります。デバイスの無線のオン/オフは、有線 USB デバイスのホットプラグに似ています。

associate サブコマンドによって作成された関連付け情報は、デバイスとホストの非揮発性メモリー内に保持されます。remove-dev または remove-host サブコマンドを使用すると、ホストからこの情報を削除できます。デバイス上の情報は、別の関連付けで上書きできます。複数のホストに関連付けられたデバイスで複数の関連付けレコードの優先順位を決定したりそれらのレコードを更新したりする方法は、メーカーごとに異なります。

各 wusbadm サブコマンドは、次のいずれかのオブジェクトに対して動作します。

host-id
: システム上のワイヤレス USB ホストを一意に識別する 2 桁の数字 (01 から 99 の範囲)。この数字は、wusb サービス (「注意事項」節を参照) が正常に有効化され、このサービスがホストインスタンスをはじめて検出したときに生成されます。この数字は、remove-host サブコマンドで削除されるまで維持されます。

dev-id
: ワイヤレス USB ホストに関連付けられたワイヤレス USB デバイスを一意に識別する 5 桁の数字。先頭の 2 桁は、デバイスが関連付けられたワイヤレス USB ホストのホスト ID です。末尾の 3 桁の数字 (001 から 999 の範囲) は、同じホストに関連付けられたデバイスを区別するために使用されます。この 5 桁の数字の先頭 2 桁と末尾 3 桁はドットで区切られます。

  dev-id は、デバイス関連付け処理中に生成されます。この値は、remove-dev サブコマンドによって削除されるか、同じホストとデバイスの間の別の関連付けによって更新されるまで、デバイスに対して維持されます。

サブコマンド

次のサブコマンドがサポートされています。list 以外のすべてのサブコマンドは、実行時にオプションやオペランドが指定されなかった場合に、サブコマンド固有の使用法に関する情報を表示します。

list [-h | -d] [-o *field*[,...]]

システム上のワイヤレス USB のホストとデバイスを一覧表示し、すべてのホストとデバイスについて ID、状態、およびタイプを表示します。list はデフォルトで、すべてのホストとデバイス、およびすべてのフィールドを一覧表示します。各ホストとそのデバイスが 1 つのグループとして表示されます。このサブコマンドは、次のオプションをサポートします。

-o *field*[,...]、 --output=*field*[,...]

表示する出力フィールドのコンマ区切りのリスト (大文字と小文字は区別されません)。フィールド名は、次に示すフィールドのいずれか、またはすべてのフィールドを表示するための特殊な値 all にする必要があります。デフォルトでは (-o の指定なし)、list はすべてのフィールドを表示します。

ID

host-id または *dev-id*。

TYPE

host タイプまたは device タイプ。

host のタイプとしては、whci (オンボードのホスト) と hwa (ホットプラグ対応ホスト) があります。

device のタイプとしては、kbd、mouse、storage、printer、dwa (ワイヤレス USB ハブ)、audio、video などがあります。

STATE

ホストには次の状態が存在します。

enabled

ホストは、関連付けの実行、デバイスの接続、データ通信の実行などの処理を行う準備が整っているか、またはすでに処理中です。

disabled

ホストはデバイスを処理する準備が整っておらず、ホストに接続されているデバイスは存在しません。disable-host サブコマンドによってホストが停止された可能性があります。または、ホストが物理的に取り外されたかドライバが切り離されたためにホストが使用不可能になった可能性があります。

disconnected

ホストがシステムに接続されていません。hwa デバイスは、システム上の USB ポートから取り外されると、この状態になります。

デバイスには次の状態が存在します。

connected
デバイスは、ホストに接続されていて開かれる準備ができているか、またはすでに開かれていて動作中です。デフォルトでは、デバイスの関連付けが完了してその無線がオンになると、デバイスはこの状態に遷移しようとします。

disconnected
デバイスは、まだホストに接続されていないか、または開かれていません。デバイスがこの状態になった理由としては、無線が範囲外になった、電源が切れている、ハードウェアの問題が発生した、などが考えられます。

-h、--host
ワイヤレス USB ホストのみを一覧表示します。

-d、--device
ワイヤレス USB デバイスのみを一覧表示します。

associate [-h *host-id*] [[-c [-f]] | -n] [-o]
関連付け処理を開始するホストを指定します。関連付けは、ワイヤレス USB デバイスとワイヤレス USB ホストとの接続が可能になるための最初の手順です。

関連付けには次の 2 つのモデルが存在します。

ケーブルによる関連付け
まずデバイスとホストを USB ケーブルで接続し、その後、このサブコマンドを実行することで、ホストはデバイスとの関連付け情報を設定します。関連付けが有効になると、同じホストとデバイスとの間の以降の接続でケーブルが不要になります。

数字による関連付け
デバイスの無線をオンにしたあと、このサブコマンドを実行してホストにデバイスの通信先を指定します。その後、ホストとデバイスの両方で短い数字が表示されます。数字の値を比較し、ホストとデバイスの両方で確認します。

関連付けが成功したら、関連付けられた USB ホストと USB デバイスのワイヤレス接続処理に進むことができます。デフォルトでは、関連付け情報は、削除または上書きされるまでホストとデバイスの両方で保持されます。

関連付けの対象となるデバイスが複数存在する場合、このサブコマンドはそのすべてを一覧表示し、ユーザーがそれらの中から選択できるようにします。このサブコマンドには、次のオプションがあります。

-h *host-id*、--host *host-id*
関連付けの対象となるホストを指定します。このオプションが指定されなかった場合、このサブコマンドは、選択可能な有効化されたホストをすべて一覧表示します。

`-c`、`--cable`
ケーブルによる関連付け処理を開始します。ワイヤレス USB デバイスをホストに接続したあと、このオプションを指定して associate サブコマンドを実行します。

`-n`、`--numeric`
数字による関連付け処理を開始します。このサブコマンドは、ホスト上とデバイス上に表示された数字を比較するようユーザーに求めます。

前述の 2 つの関連付けモデルオプション (`-n` または `-c`) のどちらも指定されなかった場合、このサブコマンドは、いずれかの関連付けモデルオプションを指定するようユーザーに求めます。

`-f`、`--force`
ケーブルによる関連付け処理を開始します。ワイヤレス USB デバイスをホストに接続したあと、このオプションを指定して associate サブコマンドを実行します。

`-o`、`--onetime`
この関連付けが 1 回かぎりの接続用であることを示します。つまり、関連付けの実行後、このデバイスが接続されたあと切断されると、その関連付け情報がホストシステムから削除されます。次回接続時には関連付けを再度実行する必要があります。

`remove-dev` [[`-d` *dev-id*] | [`-h` *host-id*]][`-f`]
ワイヤレス USB デバイスの関連付け情報をシステムから削除します。削除後は、そのホストとデバイスに対して `associate` サブコマンドが再度実行されるまで、そのデバイスとホストは接続できません。このサブコマンドには、次のオプションがあります。

`-d`、`--device=`*dev-id*
dev-id で指定されたワイヤレス USB デバイスの関連付け情報を削除します。

`-h` *host-id*、`--host=`*host-id*
host-id で指定されたホストに関連付けられたすべてのワイヤレス USB デバイスの関連付け情報を削除します。

`-f`、`--force`
確認プロンプトを表示せずに削除を実行します。デバイスがホストに接続されていた場合、このサブコマンドはその接続を強制的に切断します。

`remove-host` [`-h` *host-id*] [`-f`]
host-id やこのホストに関連付けられたすべてのデバイスの関連付け情報などのホスト情報を、システムから削除します。このサブコマンドは通常、`hwa` ドングルのような、一時的に使用されるホットプラグ対応ワイヤレス USB ホストを削除するために使用されます。このホストを再度使用可能な状態にするには、`hwa` ドングルの物理的なホットプラグを行うなどして、このホストが再度列挙されるようにします。その後、ホスト ID が更新されます。ただし、デバイスの関連付け

情報は回復しません。オンボードのホストを削除することはお勧めしません。このサブコマンドには、次のオプションがあります。

`-h` *host-id*、`--host=`*host-id*
削除する *host-id* を指定します。

`-f`、`--force`
確認プロンプトを表示せずに削除を実行します。このホストに接続されているデバイスが 1 つ以上存在する場合には、それらの接続を強制的に切断します。

`enable-host` [`-h` *host-id*]
ホストを有効状態に遷移させます。ホストはデフォルトで有効状態になります。このサブコマンドには、次のオプションがあります。

`-h` *host-id*、`--host=`*host-id*
有効化する *host-id* を指定します。

`disable-host` [`-h` *host-id*] [`-f`]
ホストを無効状態に遷移させます。*host-id* とこのホストのすべての関連付け情報が維持されます。`enable-host` サブコマンドを発行すると、このホストは有効状態に戻ります。このサブコマンドには、次のオプションがあります。

`-h` *host-id*、`--host=`*host-id*
無効化する *host-id* を指定します。

`-f`、`--force`
確認プロンプトを表示せずに無効化処理を実行します。このホストに接続されているデバイスが 1 つ以上存在する場合、このオプションはそれらの接続を強制的に切断します。

使用例

例 1　すべてのホストとデバイスの一覧表示

次のコマンドは、ワイヤレス USB のすべてのホストとデバイスを一覧表示します。

```
# wusbadm list
01      enabled         hwa
01.001  connected       mouse
01.002  connected       kbd
02      enabled         whci
02.001  connected       printer
02.002  disconnected    storage
03      disabled        hwa
03.001  disconnected    storage
03.002  disconnected    dwa
```

例 2　ケーブルによるデバイスへの関連付け

次のコマンドは、ケーブルによる関連付けのアプローチを使用して、特定のホスト (ホスト ID `01`) にデバイスを関連付けます。

例 2 ケーブルによるデバイスへの関連付け (続き)

```
# wusbadm associate -h 01 -c
Associate a device with host (01) via cable.
Continue (yes/no)?
```

例 3 デバイスの関連付けの削除

次のコマンドは、デバイスの関連付け情報をホストシステムから削除します。

```
# wusbadm remove-dev -d 01.002
Remove the information of device (01.002) from system.
This device can not be connected with the host until it is associated
again. Continue (yes/no)?
```

例 4 すべてのデバイスの関連付けの削除

次のコマンドは、特定のホストに関連付けられたすべてのデバイスの関連付け情報を削除します。

```
# wusbadm remove-dev -h 02
Remove the information of all the devices associated with host (02)
from the system.
All the devices associated with the host cannot be connected with it
until they are associated again. Continue (yes/no)?
```

終了ステータス

次の終了値が返されます。

0
: 正常終了。

1
: エラー。処理が失敗しました。たとえば、デバイスとホストの関連付けが失敗しました。

2
: 使用方法のエラー。

属性

属性についての詳細は、attributes(5) を参照してください。

/usr/sbin

属性タイプ	属性値
使用条件	system/io/usb
インタフェースの安定性	確実

関連項目

attributes(5), hwahc(7D), usba(7D)

注意事項

wusb (ワイヤレス USB の管理) サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/wusb:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスのステータスを照会するには、svcs(1) コマンドを使用します。

wusb サービスは、非公開インタフェースの wusbd デーモンによって実装されています。このデーモンは wusb サービスと同様に、SMF によって起動されます。このデーモンを指定するには、次のサービスインスタンスを使用します。

```
svc:/system/wusbd:default
```

wusbd デーモンを直接呼び出さないようにしてください。

名前

ypbind – NIS binder process

形式

```
/usr/lib/netsvc/yp/ypbind [-broadcast | -ypset | -ypsetme]
```

機能説明

NIS provides a simple network lookup service consisting of databases and processes. The databases are stored at the machine that runs an NIS server process. The programmatic interface to NIS is described in ypclnt(3NSL). Administrative tools are described in ypinit(1M), ypwhich(1), and ypset(1M). Tools to see the contents of NIS maps are described in ypcat(1), and ypmatch(1).

ypbind is a daemon process that is activated at system startup time from the svc:/network/nis/client:default service. By default, it is invoked as ypbind -broadcast. ypbind runs on all client machines that are set up to use NIS. The function of ypbind is to remember information that lets all NIS client processes on a node communicate with some NIS server process. ypbind must run on every machine which has NIS client processes. The NIS server may or may not be running on the same node, but must be running somewhere on the network.

The SMF service svc:/network/nis/client has the following properties in the config property group:

```
config.use_broadcast
config.use_ypsetme
```

The information ypbind remembers is called a *binding* — the association of a domain name with a NIS server. The process of binding is driven by client requests. As a request for an unbound domain comes in, if started with the -broadcast option, the ypbind process broadcasts on the net trying to find an NIS server, that is, a ypserv process serving the domain with a name the same as (case sensitive) the name of the domain in the client request. Since the binding is established by broadcasting, there must be at least one NIS server on the net. If started without the -broadcast option, ypbind process steps through the list of NIS servers that was created by ypinit -c for the requested domain. There must be an NIS server process on at least one of the hosts in the NIS servers file. It is recommended that you list each of these NIS servers by name and numeric IP address in /etc/hosts. Though the practice is not recommended, NIS allows you to list servers by numeric address only, bypassing /etc/hosts. In such a configuration, ypwhich(1) returns a numeric address instead of a name.

Once a domain is bound by ypbind, that same binding is given to every client process on the node. The ypbind process on the local node or a remote node may be queried for the binding of a particular domain by using the ypwhich(1) command.

If ypbind is unable to speak to the NIS server process it is bound to, it marks the domain as unbound, tells the client process that the domain is unbound, and tries to bind the domain once again. Requests received for an unbound domain will wait until the requested domain is bound. In general, a bound domain is marked as unbound when the node running the NIS server crashes or gets overloaded. In such a case, ypbind will try to bind to another NIS server using the process described above.ypbind also accepts requests to set its binding for a

particular domain. The request is usually generated by the ypset(1M) command. In order for ypset to work, ypbind must have been invoked with flags -ypset or -ypsetme.

オプション

-broadcast
: Send a broadcast datagram using UDP/IP that requests the information needed to bind to a specific NIS server. This option is analogous to ypbind with no options in earlier Sun releases and is recommended for ease of use.

  Enabling the SMF property config.use_broadcast enables -broadcast.

-ypset
: Allow users from any remote machine to change the binding by means of the ypset command. By default, no one can change the binding. This option is insecure.

-ypsetme
: Only allow root on the local machine to change the binding to a desired server by means of the ypset command. ypbind can verify the caller is indeed a root user by accepting such requests only on the loopback transport. By default, no external process can change the binding.

  Enabling the SMF property config.use_ypsetme enables -ypsetme.

ファイル

/var/yp/binding/*ypdomain*/ypservers
: Lists the servers to which the NIS client is allowed to bind.

/etc/inet/hosts
: File in which it is recommended that NIS servers be listed.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

svcs(1), ypcat(1), ypmatch(1), ypwhich(1), ifconfig(1M), svcadm(1M), ypinit(1M), ypset(1M), ypclnt(3NSL), hosts(4), ypfiles(4), attributes(5), smf(5)

注意事項

ypbind supports multiple domains. The ypbind process can maintain bindings to several domains and their servers, the default domain is the one specified by the domainname(1M) command at startup time.

The -broadcast option works only on the UDP transport. It is insecure since it trusts “any” machine on the net that responds to the broadcast request and poses itself as an NIS server.

The ypbind service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/nis/client:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using `svcadm(1M)`. The service's status can be queried using the `svcs(1)` command.

名前

ypinit – set up NIS client

形式

`/usr/sbin/ypinit [-c] [-m] [-s` *master_server*`]`

機能説明

`ypinit` can be used to set up an NIS client system. You must be the superuser to run this command. This script need not be used at all if `ypbind(1M)` is started with the `-broadcast` option (it is invoked with this option from the `svc:/network/nis/client:default service`).

Normally, `ypinit` is run only once after installing the system. It may be run whenever a new NIS server is added to the network or an existing one is decommissioned.

`ypinit` prompts for a list of NIS servers to bind the client to; this list should be ordered from the closest to the furthest server. It is recommended that you list each of these NIS servers by name and numeric IP address in `/etc/hosts`. Though the practice is not recommended, NIS allows you to list servers by numeric address only, bypassing `/etc/hosts`. In such a configuration, `ypwhich(1)` returns a numeric address instead of a name.

`ypinit` stores the list of servers to which a client can bind in the file `/var/yp/binding/`*domain*`/ypservers`. This file is used by `ypbind` when run without the `-broadcast` option.

オプション

`-c`
: Set up a `ypclient` system.

`-m`
: Build a master `ypserver` data base.

`-s` *master_server*
: Slave data base. *master_server* must be the same master configured in the YP maps and returned by the `ypwhich -m` command.

ファイル

`/etc/hosts`
: File in which it is recommended that NIS servers be listed.

`/var/yp/binding/`*domain*`/ypservers`
: Lists the servers to which the NIS client is allowed to bind.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`svcs(1)`, `ypwhich(1)`, `svcadm(1M)`, `ypbind(1M)`, `sysinfo(2)`, `hosts(4)`, `attributes(5)`, `smf(5)`

注意事項

The NIS client service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/nis/client:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

使用上の留意点

ypinit sets up the list of NIS servers only for the current domain on the system when it is run, that is, the domain returned by the SI_SRPC_DOMAIN command to sysinfo(2). Care should be taken to ensure that this is the same as the desired domain for NIS client processes.

名前

ypmake – rebuild NIS database

形式

```
cd /var/yp ; make [map]
```

機能説明

The file called `Makefile` in `/var/yp` is used by `make(1S)` to build the Network Information Service (NIS) database. With no arguments, `make` creates `dbm` databases for any NIS maps that are out-of-date, and then executes `yppush(1M)` to notify slave databases that there has been a change.

If you supply a *map* on the command line, `make` will update that map only. Typing `make passwd` will create and `yppush` the password database (assuming it is out of date). Likewise, `make ipnodes` and `make networks` will create and `yppush` the `ipnodes` and `network` files, `$(INETDIR)/ipnodes` and `$(DIR)/networks`.

There are four special variables used by `make`: `DIR`, which gives the directory of the source files; `NOPUSH`, which when non-null inhibits doing a `yppush` of the new database files; `INETDIR`, which gives the directory of the `ipnodes` source file; and `DOM`, which is used to construct a domain other than the master's default domain. The default for `DIR` is `/etc`, and the default for `INETDIR` is `/etc/inet`. The default for `NOPUSH` is the null string.

Refer to `ypfiles(4)` and `ypserv(1M)` for an overview of the NIS service.

If a NIS to LDAP (N2L) configuration file, `/var/yp/NISLDAPmapping`, is present, the NIS server components run in N2L mode. In N2L mode, the server components use a new set of map files with an LDAP-prefix, based on the LDAP DIT. In N2L mode, authoritative NIS information is obtained from the DIT. The NIS source files and `ypmake` have no role, and they should not be used. If `ypmake` is accidentally run, then the server components will detect this, and will log a warning message. For additional information, see `ypfiles(4)`.

ファイル

`/var/yp`	Directory containing NIS configuration files.
`/etc/inet/hosts`	System hosts file.
`/etc`	Default directory for source files.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis
Interface Stability	Committed

関連項目

`make(1S)`, `makedbm(1M)`, `ypbind(1M)`, `yppush(1M)`, `ypserv(1M)`, `ypclnt(3NSL)`, `NISLDAPmapping(4)`, `ypfiles(4)`, `ypserv(4)`

注意事項

The NIS makefile is only used when running the ypserv(1M) server to provide NIS services. See ypfiles(4) for more details.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same. Only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications plc, and may not be used without permission.

名前

ypmap2src – convert NIS maps to NIS source files

形式

```
/usr/lib/netsvc/yp/ypmap2src [-t]
     [ [-c custom-map-name]]... [-d domain] -o output-directory
     [ [ source-file]]...
```

機能説明

Use the `ypmap2src` utility to convert standard NIS maps to approximations of the equivalent NIS source files. This utility functions like the reverse of `ypmake(1M)`.

The primary use for `ypmap2src` is to convert from a NIS server that uses the NIS to LDAP(N2L) transition mechanism, which does not use NIS source files, to traditional NIS, where source files are required. The `ypmap2src` utility is also used by NIS administrators who wish to discover the contents of NIS maps for which the sources are not available.

Generally, this operation is not necessary. More often, administrators will switch from traditional NIS to N2L in anticipation of the eventual transition to LDAP naming. When this switch is made, authoritative information is moved into the LDAP DIT, and the NIS sources have no further role. N2L supports NIS clients until such time as they can be converted to LDAP, and the NIS service suspended.

The `ypmap2src` utility does not guarantee that the files that are generated are identical to the original NIS source files. Some information might have been thrown away by `ypmake` and cannot be recovered. N2L also might have updated the maps to reflect changes made by LDAP clients. It is essential that the sources generated are checked to confirm no problems have occurred.

Per entry comment fields, from existing source files, are not merged into source files generated by `ypmap2src`. If a user wishes N2L to maintain comment information, then the `NISLDAPmapping` configuration file should be modified so that the comment fields are mapped into LDAP. This will ensure that the comments are visible to native LDAP clients and present in the N2L map files.

When `ypmap2src` is run, it will take up-to-date comments from the map file and insert them into the NIS source file generated.

Handling Custom Maps

`ypmap2src` only knows about the standard NIS maps and standard source to map conversion. If an advanced user has changed these, that is, the user has modified the NIS makefile, the equivalent changes must also be made to the `ypmap2src` script.

オプション

`ypmap2src` supports the following options:

`-c`	Specifies that *custom-map-name* should be converted to a source file by running `makedbm -u` on it. This is a short cut so that simple custom maps can be handled without editing `ypmap2src`.
`-d` *domain-name*	Specifies the domain to convert. The *domain-name* can be a fully qualified file path, such as `/var/yp/a.b.c`, or just a domain name, a.b.c. In the latter case, `ypmaptosrc` looks in `/var/yp` for the domain directory.

-o *dest* Specifies the destination directory for the converted files. A directory other than /etc should be specified. The maps generated are copied to the correct location, /etc, /etc/security or other source directory, as appropriate.

-t Specifies that traditional NIS maps, without N2L's LDAP_ prefix, should be converted. By default, maps with the LDAP_ prefix are converted.

オペランド

ypmap2src supports the following operands:

source-file Lists the standard source files to convert. If this option is not given, then all the standard source files, plus any custom files pecified by the -c option, are converted.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis
Interface Stability	Obsolete

関連項目

ypmake(1M), ypserv(1M), NISLDAPmapping(4), attributes(5)

『Oracle Solaris Administration: Naming and Directory Services』

名前

yppoll – return current version of a NIS map at a NIS server host

形式

`/usr/sbin/yppoll [-d ypdomain] [-h host] mapname`

機能説明

The `yppoll` command asks a `ypserv( )` process what the order number is, and which host is the master NIS server for the named map.

オプション

`-d` *ypdomain* Use *ypdomain* instead of the default domain.

`-h` *host* Ask the `ypserv` process at *host* about the map parameters. If *host* is not specified, the NIS server for the local host is used. That is, the default host is the one returned by `ypwhich(1)`.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`ypwhich(1)`, `ypfiles(4)`, `attributes(5)`

名前

yppush – force propagation of changed NIS map

形式

```
/usr/lib/netsvc/yp/yppush [-v] [-h host] [-d domain]
      [-p #parallel-xfrs] mapname
```

機能説明

`yppush` copies a new version of a Network Information Service (NIS) map from the master NIS server to the slave NIS servers. It is normally run only on the master NIS server by the `Makefile` in `/var/yp` after the master databases are changed. It first constructs a list of NIS server hosts by reading the NIS `ypservers` map within the *domain*. Keys within the `ypservers` map are the ASCII names of the machines on which the NIS servers run.

A "transfer map" request is sent to the NIS server at each host, along with the information needed by the transfer agent (the program that actually moves the map) to call back the `yppush`. When the attempt has completed (successfully or not), and the transfer agent has sent `yppush` a status message, the results can be printed to `stdout`. Messages are also printed when a transfer is not possible, for instance, when the request message is undeliverable, or when the timeout period on responses has expired.

Refer to `ypfiles(4)` and `ypserv(1M)` for an overview of the NIS service.

オプション

The following options are supported:

`-d` *domain*	Specifies a *domain*.
`-h` *host*	Propagates only to the named *host*.
`-p` *#parallel-xfrs*	Allows the specified number of map transfers to occur in parallel.
`-v`	Verbose. This prints messages when each server is called, and for each response. If this flag is omitted, only error messages are printed.

ファイル

`/var/yp`	Directory where NIS configuration files reside.
`/var/yp/`*domain*`/ypservers.` {*dir*, *pag*}	Map containing list of NIS servers to bind to when running in server mode.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis

関連項目

`ypserv(1M)`, `ypxfr(1M)`, `ypfiles(4)`, `attributes(5)`

注意事項

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications PLC, and must not be used without permission.

使用上の留意点

In the current implementation (version 2 NIS protocol), the transfer agent is ypxfr(1M), which is started by the ypserv program. If yppush detects that it is speaking to a version 1 NIS protocol server, it uses the older protocol, sending a version 1 YPPROC_GET request and issues a message to that effect. Unfortunately, there is no way of knowing if or when the map transfer is performed for version 1 servers. yppush prints a message saying that an “old-style” message has been sent. The system administrator should later check to see that the transfer has actually taken place.

名前

ypserv, ypxfrd – NIS server and binder processes

形式

```
/usr/lib/netsvc/yp/ypserv [-dv] [-i | -I] [-r | -R]

/usr/lib/netsvc/yp/ypxfrd
```

機能説明

The Network Information Service (NIS) provides a simple network lookup service consisting of databases and processes. The databases are ndbm files in a directory tree rooted at /var/yp. See ndbm(3C). These files are described in ypfiles(4). The processes are /usr/lib/netsvc/yp/ypserv, the NIS database lookup server, and /usr/lib/netsvc/yp/ypbind, the NIS binder. The programmatic interface to the NIS service is described in ypclnt(3NSL). Administrative tools are described in yppoll(1M), yppush(1M), ypset(1M), ypxfr(1M), and ypwhich(1). Tools to see the contents of NIS maps are described in ypcat(1), and ypmatch(1). Database generation and maintenance tools are described in ypinit(1M), ypmake(1M), and makedbm(1M).

The ypserv utility is a daemon process typically activated at system startup from svc:/network/nis/server:default. Alternatively, you can, as the root user, start NIS services using ypstart(1M) from the command-line. ypserv runs only on NIS server machines with a complete NIS database. You can halt all NIS services using the ypstop(1M) command.

The ypxfrd utility transfers entire NIS maps in an efficient manner. For systems that use this daemon, map transfers are 10 to 100 times faster, depending on the map. To use this daemon, be sure ypxfrd is running on the master server. See /usr/lib/netsvc/yp/ypstart. ypxfr attempts to use ypxfrd first. If that fails, it prints a warning, then uses the older transfer method.

The ypserv daemon's primary function is to look up information in its local database of NIS maps.

The operations performed by ypserv are defined for the implementor by the *YP Protocol Specification*, and for the programmer by the header file <rpcsvc/yp_prot.h>.

Communication to and from ypserv is by means of RPC calls. Lookup functions are described in ypclnt(3NSL), and are supplied as C-callable functions in the libnsl(3LIB) library. There are four lookup functions, all of which are performed on a specified map within some NIS domain: yp_match(3NSL), yp_first(3NSL), yp_next(3NSL), and yp_all(3NSL). The yp_match operation takes a key, and returns the associated value. The yp_first operation returns the first key-value pair from the map, and yp_next can be used to enumerate the remainder. yp_all ships the entire map to the requester as the response to a single RPC request.

The SMF service svc:/network/nis/server manages the configuration of the ypserv daemon.

A number of special keys in the DBM files can alter the way in which ypserv operates. The keys of interest are:

YP_INTERDOMAIN	The presence of this key causes ypserv to forward to a DNS server host lookups that cannot be satisfied by the DBM files.
YP_SECURE	This key causes ypserv to answer only questions coming from clients on reserved ports.
YP_MULTI_*hostname*	This is a special key in the form, YP_MULTI_*hostname addr1,...,addrN*. A client looking for *hostname* receives the closest address.

Two other functions supply information about the map, rather than map entries: yp_order(3NSL), and yp_master(3NSL). In fact, both order number and master name exist in the map as key-value pairs, but the server will not return either through the normal lookup functions. If you examine the map with makedbm(1M), however, they are visible. Other functions are used within the NIS service subsystem itself, and are not of general interest to NIS clients. These functions include do_you_serve_this_domain?, transfer_map, and reinitialize_internal_state.

On start up, ypserv checks for the existence of the NIS to LDAP (N2L) configuration file /var/yp/NISLDAPmapping. If it is present then a master server starts in N2L mode. If the file is not present it starts in "traditional" (non N2L) mode. Slave servers always start in traditional mode.

In N2L mode, a new set of map files, with an LDAP_ prefix, are generated, based on the contents of the LDAP DIT. The old map files, NIS source files and ypmake(1M) are not used.

It is possible that ypmake(1M) can be accidentally run in N2L mode. If the occurs, the old style map files are overwritten. That the map files are overwritten is harmless. However, any resulting yppush(1M) operation will push information based on the DIT rather than the source files. The user may not expect information based on the DIT. ypserv keeps track of the last modification date of the old style map files. If the map files have been updated, a warning is logged that suggests that the user call yppush directly instead of ypmake.

If a server attempts to run in N2L mode and a LDAP server cannot be contacted, it behaves as follows:

1. When ypserv is started, a warning will be logged.
2. When a NIS read access is made and the TTL entry has expired, a warning is logged.Information that is returned from the cache has not been updated.
3. When a NIS write access is made, a warning is logged. The cache will not be updated, and a NIS failure will be returned.

If ypxfrd is running in N2L mode and is asked to transfer a map, ypxfrd first checks whether the map is out of date. If the map is out of date, ypxfrd initiates an update from the DIT. ypxfrd cannot wait for the update to complete. If ypxfrd waited, the client end ypxfr

operation could time out. To prevent ypxfrd from timing out, the existing map is transferred from the cache. The most up to date map will be transferred on subsequent ypxfrd operations.

オプション

ypserv

-d The NIS service should go to the DNS for more host information. This requires the existence of a correct /etc/resolv.conf file pointing to a DNS server. This option turns on DNS forwarding regardless of whether or not the YP_INTERDOMAIN flag is set in the hosts maps. See makedbm(1M). In the absence of an /etc/resolv.conf file, ypserv complains, but ignores the -d option.

If enabled, the property group/property config.service_dns tells ypserv to enable the -d option.

-i If in N2L mode, initialize the NIS related parts of the DIT based on the current, non LDAP_ prefixed, map files. The LDAP_ prefixed maps are not created or updated. If you require that LDAP_ prefixed maps be updated or created, then use the -ir option.

The -i option does not attempt to create any NIS domain or container objects. If any NIS domain or container objects have not already been created, then errors will occur, as entries are written to nonexistent containers.

-I Identical to -i, except that any missing domain and container objects are created.

-r If in N2L mode, then refresh the LDAP_ prefixed map files based on the contents of the DIT.

-ir If both -i and -r are specified in N2L mode, then the DIT will first be initialized from the current non LDAP_ prefixed map files. A new set of LDAP_ prefixed maps will then be generated from the contents of the DIT. A new set of LDAP_ prefixed maps is required when moving from traditional NIS to N2L mode NIS.

-Ir Identical to -ir, except that any missing domain and container objects are created.

-v Operate in the verbose mode, printing diagnostic messages to stderr.

When run with the -i, -r, -I, -ir or -Ir options, the ypserv command runs in the foreground and exits once map initialization has been completed. Once the ypserv command exits, the user knows the maps are ready and can restart ypserv and the other yp daemons by running ypstart(1M).

If there is a requirement to initialize the DIT from the NIS source files, which may have been modified since the maps were last remade, run ypmake before running ypserv -i or ypserv -ir. ypmake regenerated old style NIS maps. Then ypserv -ir dumps them into the DIT. When the -ir option is used, the LDAP_ prefixe maps are also generated or updated. Since these maps will be more recent than the old style maps, ypmake will not be reported as erroneous when it is run.

ファイル

/var/yp/securenets	Defines the hosts and networks that are granted access to information in the served domain. It is read at startup time by both ypserv and ypxfrd.
/var/yp/ypserv.log	If the /var/yp/ypserv.log file exists when ypserv starts up, log information is written to it when error conditions arise.
/var/yp/binding/domainname/ypservers	Lists the NIS server hosts that ypbind can bind to.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis

関連項目

svcs(1), ypcat(1), ypmatch(1), ypwhich(1), domainname(1M), makedbm(1M), svcadm(1M), ypbind(1M), ypinit(1M), ypmake(1M), yppoll(1M), yppush(1M), ypset(1M), ypstart(1M), ypstop(1M), ypxfr(1M), ndbm(3C), ypclnt(3NSL), libnsl(3LIB), NISLDAPmapping(4), securenets(4), ypfiles(4), ypserv(4), attributes(5), smf(5)

『Network Interface Guide』

『Oracle Solaris Administration: Naming and Directory Services』

注意事項

ypserv supports multiple domains. The ypserv process determines the domains it serves by looking for directories of the same name in the directory /var/yp. It replies to all broadcasts requesting yp service for that domain.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications PLC, and must not be used without permission.

NIS uses ndbm() files to store maps. Therefore, it is subject to the 1024 byte limitations described in the USAGE and NOTES sections of the ndbm(3C) man page.

The NIS server service is managed by the service management facility, smf(5), under the service identifier:

```
svc:/network/nis/server:default
```

Administrative actions on this service, such as enabling, disabling, or requesting restart, can be performed using svcadm(1M). The service's status can be queried using the svcs(1) command.

名前

ypserv_resolv – DNS forwarding service daemon for NIS

形式

`ypserv_resolv [-v | -V] [-F [-C fd]] [-t xx] [-p yy]`

機能説明

`ypserv_resolv` is an auxiliary process that provides DNS forwarding service for NIS hosts requests to `ypserv` that is running in the NIS compatibility mode. It is generally started by invoking `ypserv(1M)` with the `-d` option. Although it is not recommended, `ypserv_resolv` can also be started independently with the options shown below.

This command requires that the `/etc/resolv.conf` file be setup for communication with a DNS nameserver. The `dig(1M)` utility can be used to verify communication with a DNS nameserver. See `resolv.conf(4)` and `dig(1M)`.

オプション

`-F`	Run in foreground.
`-C` *fd*	Use *fd* for service xprt.
`-v`	Verbose. Send output to the syslog daemon.
`-V`	Verbose. Send output to stdout.
`-t` *xx*	Use transport *xx*.
`-p` *yy*	Use transient program# *yy*.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目

`dig(1M)`, `resolv.conf(4)`, `attributes(5)`

名前

ypset – point ypbind at a particular server

形式

`/usr/sbin/ypset [-d ypdomain] [-h host] server`

機能説明

In order to run `ypset`, `ypbind` must be initiated with the –`ypset` or –`ypsetme` options. See `ypbind(1M)`. `ypset` tells `ypbind` to get NIS services for the specified *ypdomain* from the `ypserv` process running on *server*. If *server* is down, or is not running `ypserv`, this might not be discovered until an NIS client process tries to obtain a binding for the domain. At this point, the binding set by `ypset` is tested by `ypbind`. If the binding is invalid, `ypbind` attempts to rebind for the same domain.

`ypset` is useful for binding a client node that is not on a broadcast net, or is on a broadcast net that is not running an NIS server host. It is also useful for debugging NIS client applications, for instance, where an NIS map exists only at a single NIS server host.

Where several hosts on the local net are supplying NIS services, `ypbind` can rebind to another host, even while you attempt to find out if the `ypset` operation succeeded. For example, if you enter the `ypset` command below, you might get the subsequent response from `ypwhich`:

```
example% ypset host1
example% ypwhich
host2
```

The sequence shown above is a function of the NIS subsystem's attempt to load-balance among the available NIS servers, and occurs when `host1` does not respond to `ypbind` because it is not running `ypserv` (or is overloaded), and `host2`, running `ypserv`, obtains the binding.

server indicates which NIS server to bind to, and must be specified as a name or an IP address. This works only if the node has a current valid binding for the domain in question and `ypbind` has been set to allow use of `ypset`. In most cases, *server* should be specified as an IP address.

`ypset` tries to bind over a connectionless transport. The NIS library call, `yp_all()`, uses connection-oriented transport and derives the NIS server's address based on the connectionless address supplied by `ypset`.

Refer to `ypfiles(4)` for an overview of the NIS name service.

オプション

`-d` *ypdomain* Use *ypdomain*, instead of the default domain.

`-h` *host* Set `ypbind`'s binding on *host*, instead of locally. Specify *host* as a name.

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

関連項目 ypwhich(1), ypfiles(4), attributes(5)

名前

ypstart, ypstop – Start and stop NIS services

形式

```
/usr/lib/netsvc/yp/ypstart
```

```
/usr/lib/netsvc/yp/ypstop
```

機能説明

The `ypstart` command is used to start the Network Information Service (NIS). After the host has been configured using the `ypinit(1M)` command, `ypstart` automatically determines the NIS status of the machine and starts the appropriate daemons.

The `ypstop` command is used to stop the Network Information Service (NIS).

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis

関連項目

`ypinit(1M)`, `attributes(5)`

『Oracle Solaris の管理: 一般的なタスク』

注意事項

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two services remains the same; only the name has changed. The name Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications PLC, and must not be used without permission.

名前

ypxfr, ypxfr_1perday, ypxfr_1perhour, ypxfr_2perday – transfer NIS map from a NIS server to host

形式

```
/usr/lib/netsvc/yp/ypxfr [-c] [-f] [-C tid prog server]
     [-d ypdomain] [-h host] [-s ypdomain] mapname
```

機能説明

The `ypxfr` command moves an NIS map in the default domain for the local host to the local host by making use of normal NIS services. It creates a temporary map in the directory `/var/yp/`*ypdomain* (this directory must already exist; *ypdomain* is the default domain for the local host), fills it by enumerating the map's entries, fetches the map parameters (master and order number), and loads them. It then deletes any old versions of the map and moves the temporary map to the real *name*.

If run interactively, `ypxfr` writes its output to the terminal. However, if it is started without a controlling terminal, and if the log file `/var/yp/ypxfr.log` exists, it appends all its output to that file. Since `ypxfr` is most often run from the privileged user's `crontab` file, or by `ypserv`, the log file can retain a record of what was attempted, and what the results were.

For consistency between servers, `ypxfr` should be run periodically for every map in the NIS data base. Different maps change at different rates: a map might not change for months at a time, for instance, and can therefore be checked only once a day. Some maps might change several times per day. In such a case, you might want to check hourly for updates. A `crontab(1)` entry can be used to automatically perform periodic updates. Rather than having a separate `crontab` entry for each map, you can group commands to update several maps in a shell script. Examples (mnemonically named) are in `/usr/sbin/yp`: `ypxfr_1perday`, `ypxfr_2perday`, and `ypxfr_1perhour`.

Refer to `ypfiles(4)` for an overview of the NIS name service.

オプション

`-c`	Do not send a “Clear current map” request to the local `ypserv` process. Use this flag if `ypserv` is not running locally at the time you are running `ypxfr`. Otherwise, `ypxfr` complains that it cannot communicate with the local `ypserv`, and the transfer fails.
`-f`	Force the transfer to occur even if the version at the master is not more recent than the local version.
`-C` *tid prog server*	This option is for use *only* by `ypserv`. When `ypserv` starts `ypxfr`, it specifies that `ypxfr` should call back a `yppush` process at the host *server*, registered as program number *prog*, and waiting for a response to transaction *tid*.
`-d` *ypdomain*	Specify a domain other than the default domain.
`-h` *host*	Get the map from *host*, regardless of the master. If *host* is not specified, `ypxfr` asks the NIS service for the name of the master, and tries to get the map from there. *host* must be a valid host name.

`-s` *ypdomain*	Specify a source domain from which to transfer a map that should be the same across domains.

ファイル

`/var/yp/ypxfr.log`	Log file
`/usr/lib/netsvc/yp/ypxfr_1perday`	Script to run one transfer per day, for use with `cron`(1M)
`/usr/lib/netsvc/yp/ypxfr_2perday`	Script to run two transfer per day, for use with `cron`(1M)
`/usr/lib/netsvc/yp/ypxfr_1perhour`	Script for hourly transfers of volatile maps
`/var/yp/`*ypdomain*	NIS domain
`/usr/spool/cron/crontabs/root`	Privileged user's `crontab` file

属性

See `attributes`(5) for descriptions of the following attributes:

ypxfr Only

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

ypxfr_1perday, ypxfr_1perhour, and ypxfr_2perday

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/nis

関連項目

`crontab`(1), `cron`(1M), `ypinit`(1M), `yppush`(1M), `ypserv`(1M), `ypfiles`(4), `attributes`(5)

名前

zdb – ZFS デバッガ

形式

`zdb` *pool*

機能説明

`zdb` コマンドは、サポートエンジニアが障害の診断や統計の収集を行うために使用します。ZFS ファイルシステムは常にディスク上での整合性が保たれるほか、自己修復機能も備えているため、`zdb` はサポートエンジニアの指示がないかぎり実行すべきではありません。

引数が指定されなかった場合、`zdb` は、プールおよび関連するデータセットに対して基本的な整合性検査を実行し、検出されたすべての問題を報告します。

このコマンドでサポートされるオプションはすべて Sun の内部用であり、いつでも変更される可能性があります。

終了ステータス

次の終了値が返されます。

`0` プールの整合性がとれています。

`1` エラーが検出されました。

`2` 無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	不確実

関連項目

`zfs(1M)`, `zpool(1M)`, `attributes(5)`

名前

zdump – time zone dumper

形式

`zdump [--version] [-v] [-c [loyear,]hiyear] [zonename]...`

機能説明

The `zdump` command prints the current time for each time zone (*zonename*) listed on the command line. Specify *zonename* as the name of the time zone database file relative to `/usr/share/lib/zoneinfo`.

Specifying an invalid time zone (*zonename*) to `zdump` does not return an error, rather `zdump` uses UTC. This is consistent with the behavior of the library calls; `zdump` reflects the same behavior of the time routines in `libc`. See `ctime(3C)` and `mktime(3C)`.

オプション

The following options are supported:

`--version` Outputs version information and exits.

`-v` Displays the entire contents of the time zone database file for *zonename*. Prints the time at the lowest possible time value; the time one day after the lowest possible time value; the times both one second before and exactly at each time at which the rules for computing local time change; the time at the highest possible time value; and the time at one day less than the highest possible time value. See `mktime(3C)` and `ctime(3C)` for information regarding time value (`time_t`). Each line of output ends with `isdst=1` if the given time is Daylight Saving Time, or `isdst=0` otherwise.

`-c` [*loyear*],*hiyear* Cuts off the verbose output near the start of the given year(s) . By default, the program cuts off verbose output near the start of the years -500 and 2500.

終了ステータス

The following exit values are returned:

`0` Successful completion.

`1` An error occurred.

ファイル

`/usr/share/lib/zoneinfo` Standard zone information directory

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

関連項目

`zic(1M)`, `ctime(3C)`, `mktime(3C)`, `attributes(5)`, `environ(5)`

名前

zfs – ZFS ファイルシステムの構成

形式

```
zfs [-?]

zfs help subcommand | help | property property-name | permission

zfs create [-p] [-o property=value] ... filesystem

zfs create [-ps] [-b blocksize] [-o property=value] ... -V size volume

zfs destroy [-rRf] filesystem|volume

zfs destroy [-rRd] snapshot

zfs snapshot [-r] [-o property=value]...
      filesystem@snapname|volume@snapname

zfs rollback [-rRf] snapshot

zfs clone [-p] [-K] [-o property=value] ... snapshot filesystem|volume

zfs promote clone-filesystem

zfs rename filesystem|volume|snapshot
     filesystem|volume|snapshot

zfs rename [-p] filesystem|volume filesystem|volume

zfs rename -r snapshot snapshot

zfs list [-rH|-d depth][-o property[,...]] [-t type[,...]]
     [-s property] ... [-S property] ...
     [filesystem|volume|snapshot|path] ...

zfs set [-r] property=value filesystem|volume|snapshot ...

zfs get [-rHp|-d depth][-o all | field[,...]] [-s source[,...]]
     all | property[,...] filesystem|volume|snapshot ...

zfs set share=name=sharename,path=pathname,[desc=description],
     prot=smb | nfs[,property=value[,...]] filesystem

zfs set -c share=name=sharename filesystem

zfs get share [filesystem]

zfs inherit [-rS] property filesystem|volume|snapshot ...

zfs upgrade

zfs upgrade [-v]

zfs upgrade [-r] [-V version] -a | filesystem

zfs userspace [-hniHp] [-o field[,...]] [-sS field] ...
     [-t type [,...]] filesystem|snapshot

zfs groupspace [-hniHpU] [-o field[,...]] [-sS field] ...
     [-t type [,...]] filesystem|snapshot
```

```
zfs mount

zfs mount [-vOS] [-o options] -a | filesystem

zfs unmount [-f] -a | filesystem|mountpoint

zfs share -a | dataset

zfs unshare -a dataset|mountpoint

zfs send [-DvRbp] [-[iI] snapshot] snapshot

zfs send -r [-Dvbcp] [-i snapshot] snapshot

zfs receive [-vnFu] [[-o property=value] | [-x property]] ...
     filesystem|volume|snapshot

zfs receive [-vnFu] [[-o property=value] | [-x property]] ...
     [-d | -e] filesystem

zfs allow filesystem|volume

zfs allow [-ldug] everyone|user|group[,...] perm|@setname[,...]
     filesystem|volume

zfs allow [-ld] -e perm|@setname[,...] filesystem|volume

zfs allow -c perm|@setname[,...] filesystem|volume

zfs allow -s @setname perm|@setname[,...] filesystem|volume

zfs unallow [-rldug] everyone|user|group[,...] [perm|@setname[,... ]]
     filesystem|volume

zfs unallow [-rld] -e [perm|@setname[,... ]] filesystem|volume

zfs unallow [-r] -c [perm|@setname[ ... ]] filesystem|volume

zfs unallow [-r] -s @setname [perm|@setname[,... ]] filesystem|volume

zfs hold [-r] tag snapshot...

zfs holds [-r] snapshot...

zfs release [-r] tag snapshot...

zfs diff [-FHte] [-o field] ... snapshot [snapshot|filesystem]

zfs diff -E [-FHt] [-o field] ... snapshot|filesystem

zfs key -l {-a | [-r] filesystem|volume}

zfs key -u [-f] {-a | [-r] filesystem|volume}

zfs key -c [-o keysource=value] {-a | [-r] filesystem|volume}

zfs key -K {-a | [-r] filesystem|volume}
```

機能説明

zfs コマンドは、ZFS ストレージプール内の ZFS データセットを構成します。詳細は、zpool(1M) を参照してください。データセットの識別には、ZFS 名前空間内部の一意のパスが使用されます。例:

pool/{*filesystem*,*volume*,*snapshot*}

この場合、データセット名の最大長は MAXNAMELEN (256 バイト) です。

使用できるデータセットは、次のうちのいずれかです。

file system
: 標準のシステム名前空間内にマウントでき、別のファイルシステムのように動作する、filesystem タイプの ZFS データセット。ZFS は POSIX 準拠のファイルシステムとして設計されていますが、既知の問題により、準拠の要件が満たされない場合がいくつかあります。標準準拠に依存しているアプリケーションでは、ファイルシステムの空き領域を調べるときに、標準でない動作のために障害が発生する可能性があります。

volume
: raw デバイスまたはブロック型デバイスとしてエクスポートされた論理ボリューム。このタイプのデータセットは特殊な状況でのみ使用するようにしてください。通常は、ほとんどの環境でファイルシステムが使用されます。

snapshot
: 特定の時点におけるファイルシステムまたはボリュームの読み取り専用バージョン。これは、*filesystem@name* または *volume@name* として指定されます。

ZFS ファイルシステム階層

ZFS ストレージプールは、データセット用の領域を提供するデバイスの論理コレクションです。ストレージプールは、ZFS ファイルシステム階層のルートでもあります。

プールのルートにはファイルシステムとしてアクセスできるため、マウントやマウント解除、スナップショットの作成、およびプロパティーの設定などの操作を実行できます。ただし、物理ストレージ特性の管理には、zpool(1M) コマンドを使用します。

プールの作成および管理の詳細は、zpool(1M) を参照してください。

スナップショット

スナップショットは、ファイルシステムまたはボリュームの読み取り専用コピーです。スナップショットはきわめてすばやく作成することができ、最初はプール内で追加の領域を消費しません。アクティブなデータセット内のデータが変更されると、スナップショット用のデータが増加します。これは、本来ならアクティブなデータセットと共有されていたデータです。

スナップショットには、任意の名前を付けることができます。ボリュームのスナップショットは、クローンやロールバックを作成することは可能ですが、個別にアクセスすることはできません。

ファイルシステムのスナップショットは、ファイルシステムのルートの .zfs/snapshot ディレクトリに置かれます。スナップショットは、要求に応じて自動的にマウントされます。また、一定の時間が経過するとスナップショットをマウント解除することもできます。.zfs ディレクトリの可視設定は、snapdir プロパティーで制御できます。

クローン

クローンとは、別のデータセットと同一の初期内容を持つ書き込み可能なボリュームまたはファイルシステムです。スナップショットの場合と同様に、クローンは瞬間的に作成され、最初は追加の領域を消費しません。

クローンは、スナップショットからのみ作成できます。スナップショットのクローンを作成すると、親子間に暗黙の依存関係が作成されます。クローンをデータセット階層内の別の場所に作成した場合でも、クローンが存在するかぎり、元のスナップショットを破棄することはできません。この依存関係は、origin プロパティーからわかります。それらが存在する場合には、 destroy コマンドを実行すると表示されます。

promote サブコマンドを使えば、クローンの親子依存関係を逆転させることができます。これを行うと、「元」のファイルシステムが、指定されたファイルシステムのクローンになります。このため、クローンの作成元のファイルシステムを破棄できるようになります。

マウントポイント

ZFS ファイルシステムは簡単な操作で作成できるため、システムごとのファイルシステムの数はかなり多くなる可能性があります。この状況に対応するため、/etc/vfstab ファイルを編集しなくても、ZFS はファイルシステムのマウントおよびマウント解除を自動的に管理します。自動管理されるすべてのファイルシステムは、ブート時に ZFS によりマウントされます。

デフォルトでは、ファイルシステムは */path* 以下にマウントされます。ここで、*path* は ZFS 名前空間内のファイルシステムの名前です。ディレクトリが、必要に応じて作成および破棄されます。

また、mountpoint プロパティーにファイルシステムのマウントポイントを設定することも可能です。このディレクトリは必要に応じて作成され、zfs mount -a コマンドの呼び出し時に ZFS によりファイルシステムが自動的にマウントされます (/etc/vfstab の編集は不要)。この mountpoint プロパティーは継承可能です。このため、pool/home が /export/stuff のマウントポイントを保持する場合、pool/home/user は自動的に /export/stuff/user のマウントポイントを継承します。

zfs mount コマンドに -o mountpoint=*value* オプションを指定すると、ファイルシステムの永続的なマウントポイント以外の場所に、ファイルシステムを一時的にマウントできます。これは、レガシーマウントポイントを持つファイルシステムで可能です。

ファイルシステムの mountpoint プロパティーが none の場合、ファイルシステムはマウントされません。

必要に応じ、従来のツール (mount、umount、/etc/vfstab) を使って ZFS ファイルシステムを管理することもできます。ファイルシステムのマウントポイントが legacy に設定されている場合、ZFS はファイルシステムの管理を試みません。管理者が、ファイルシステムのマウントおよびマウント解除を担当します。

ゾーン

追加したファイルシステムの物理プロパティーの制御は、大域管理者が行います。ただし、ゾーン管理者は、追加したファイルシステム内部のファイルを、ファイルシステムのマウント方法に対応した方法で作成、変更、または破棄できます。

データセットを非大域ゾーンに委任する場合は、zonecfg add dataset サブコマンドを使用します。データセットをあるゾーンに委任し、同じデータセットの子を別のゾーンに委任することはできません。ゾーン管理者は、データセットおよびそのすべての子のプロパティーを変更できます。ただし、quota プロパティーの制御は、大域管理者が行います。

ZFS ボリュームをデバイスとして非大域ゾーンに追加するには、zonecfg add device サブコマンドを使用します。ただし、その物理プロパティーを変更できるのは大域管理者だけです。

zonecfg の構文の詳細は、zonecfg(1M) を参照してください。

データセットが非大域ゾーンに委任されると、zoned プロパティーが自動的に設定されます。ゾーン設定したファイルシステムは、一時的な mountpoint プロパティー (「一時的なマウントポイントプロパティー」を参照) を使用することによってのみ、大域ゾーンでマウントできます。

大域管理者は zoned プロパティーを強制的にクリアできます。ただし、この操作は特に慎重に行うようにしてください。大域管理者は、このプロパティーをクリアする前に、すべてのマウントポイントが受け入れられるものであることを確認してください。

複製解除

複製解除とは、冗長なデータをブロックレベルで削除することにより、格納されているデータの合計容量を減らすプロセスです。複製解除はプール全体に及びますが、データセットごとにそれぞれの dedup プロパティーを使用すると、適用するかしないかを選択できます。ファイルシステムで dedup プロパティーが有効になっている場合、重複しているデータブロックは書き込み時に一斉に削除されます。この結果、ただ 1 つの一意のデータが格納され、共通するコンポーネントが、プール内の dedup が有効になっているすべてのデータセットのファイル間で共有されます。

暗号化

ZFS 暗号化および ZFS 暗号化構文についての完全な説明は、zfs_encrypt(1M) を参照してください。

ネイティブプロパティー

プロパティーは、ネイティブプロパティーとユーザー定義 (ユーザー) プロパティーの 2 種類に分けられます。ネイティブプロパティーは、内部統計情報を提供するか、ZFS の動作を制御します。また、ネイティブプロパティーは編集可能であるか、読み取り専用です。ユーザープロパティーは ZFS の動作には影響を及ぼしま

せんが、ユーザーの環境で意味のある方法でデータセットに注釈を付けるのに利用できます。ユーザープロパティーの詳細は、後述の「ユーザープロパティー」の節を参照してください。

すべてのデータセットは、データセットに関する統計情報を提供したり、さまざまな動作を制御したりするプロパティーセットを持っています。プロパティーは、子により上書きされないかぎり、親から継承されます。一部のプロパティーは、特定のタイプのデータセット (ファイルシステム、ボリューム、またはスナップショット) にのみ適用されます。

数値プロパティーの値は、人間が読み取ることのできるサフィックス (k、KB、M、Gb など) を使用すると指定できます。使用可能な最大サフィックスは Z (z バイト) です。次の指定はすべて有効 (および等価) です。

```
1536M, 1.5g, 1.50GB
```

数値以外のプロパティーの値は、大文字と小文字が区別されるため、mountpoint プロパティーを除いて小文字にする必要があります。

次のネイティブプロパティーは、データセットに関する統計データを表示するものです。これらのプロパティーは、設定することも継承することもできません。ネイティブプロパティーは、特に注記がないかぎり、すべてのデータセットタイプに適用されます。

available
: データセットおよびその子すべてが使用可能な容量。プール内にその他のアクティビティーが存在しないものとして計算されます。容量はプール内で共有されるため、プールの物理サイズ、割り当て、予約、プール内のほかのデータセットなどのさまざまな要因によって、利用できる容量が制限されることがあります。

  このプロパティーは、列の短縮名 avail を使用して参照することもできます。

compressratio
: このデータセットに対して達成された圧縮比。乗数で表記されます。zfs set compression=on *dataset* を実行すると、圧縮を有効にできます。デフォルト値は off です。

creation
: このデータセットが作成された時刻。

defer_destroy
: zfs destroy -d コマンドを使ってスナップショットの遅延破棄が予約されている場合、プロパティーがオンになります。それ以外の場合、このプロパティーはオフです。

keystatus
: 詳細は、zfs_encrypt(1M) を参照してください。

mounted
ファイルシステムの場合は、ファイルシステムが現在マウントされているかどうかを示します。このプロパティーは、yes または no になります。

origin
ファイルシステムまたはボリュームのクローンを作成した場合は、クローンの作成元のスナップショット。クローンが存在するかぎり、-r や -f オプションを使用しても、作成元は破棄できません。

referenced
このデータセットでアクセス可能なデータ量。これは、プール内のほかのデータセットと共有される場合も、共有されない場合もあります。スナップショットまたはクローンを作成したときには、それらの作成元のファイルシステムまたはスナップショットと同じ領域を最初は参照しています。内容が同じであるためです。

このプロパティーは、列の短縮名 refer を使用して参照することもできます。

rekeydate
詳細は、zfs_encrypt(1M) を参照してください。

type
データセットのタイプ。filesystem、volume、snapshot のいずれかになります。

used
このデータセットおよびそのすべての子孫が消費する容量を調べます。この値は、このデータセットの割り当ておよび予約に基づいて計算されます。使用されている容量には、このデータセットの予約は含まれませんが、(usedbyrefreservation による) refreservation 、および (usedbychildren による) 子孫データセットの予約が考慮されます。データセットがその親から継承して使用する容量、およびこのデータセットが再帰的に破棄されるときに解放される容量は、使用済み領域と予約のどちらか大きい方になります。

スナップショット (「スナップショット」の節を参照) を作成したときは、それらの領域は最初はスナップショットとファイルシステムの間で共有されます。それまでに作成したスナップショットと領域が共有されることもあります。ファイルシステムが変化していくにつれて、それまで共有されていた領域がスナップショット固有になり、スナップショットが使用する領域に計上されます。また、スナップショットを削除すると、ほかのスナップショットに固有の (および使用される) 容量を増やすことができます。

使用している容量、使用できる容量、または参照する容量では、保留状態の変更は考慮されません。保留状態の変更は通常、数秒以内に計上されます。fsync(3C) や O_SYNC を使用してディスクへの変更をコミットしても、領域の使用状況の情報がすぐに更新されることが保証されているわけではありません。

usedby*
usedby* プロパティーは、used プロパティーをその領域のさまざまな使用理由に分解します。具体的には、used = usedbychildren + usedbydataset + usedbyrefreservation + usedbysnapshots となります。これらのプロパティーは、「version 13」またはそれ以上のプールで作成されたデータセットでのみ使用可能です。

usedbychildren
このデータセットの子によって使用される容量。この容量は、データセットのすべての子が破棄された場合に解放されます。

usedbydataset
このデータセット自体によって使用される容量。この容量は、データセットが破棄された場合に解放されます (まず refreservation をすべて削除し、必要に応じてスナップショットや子孫を破棄したあと)。

usedbyrefreservation
このデータセットの refreservation セットによって使用される容量。この容量は、refreservation が削除された場合に解放されます。

このプロパティーで計上される容量は、将来の書き込みによって見込まれる消費量を表しており、このデータセット内での書き込み割り当ての失敗を防止するためにあらかじめ予約されています。これには、未書き込みデータ、現在スナップショットと共有されている容量、および (圧縮率の低いデータと置き換えると失われる可能性がある) ボリュームの圧縮保存が含まれます。将来の書き込み用に容量を割り当てると、usedbydataset または usedbysnapshots が増え、それに応じて usedbyrefreservation が減ります。

usedbysnapshots
このデータセットのスナップショットによって消費される容量。具体的には、これはデータセットのスナップショットがすべて破棄された場合に解放される容量です。ただし、これは、スナップショットの used プロパティーの単なる合計ではありません。なぜならば、領域は、複数のスナップショットによって共有されることがあるからです。

userused@*user*
指定されたユーザーによって消費される、このデータセットの容量。容量は、ls -l で表示される各ファイルの所有者にチャージされます。チャージされた容量は du および ls -s で表示されます。詳細は、zfs userspace サブコマンドを参照してください。

非特権ユーザーは、自身の領域使用状況のみを確認できます。root ユーザー、または zfs allow で userused 特権を付与されているユーザーは、あらゆるユーザーの使用状況にアクセスできます。

userused@... プロパティーは zfs get all では表示されません。@ 記号のあとに、次のいずれかの形式でユーザーの名前を付加する必要があります。

- *POSIX* 名 (例: joe)

- *POSIX* 数値 *ID* (例: 789)
- *SID* 名 (例: joe.smith@mydomain)
- *SID* 数値 *ID* (例: S-1-123-456-789)

userrefs
このプロパティーは、このスナップショットでユーザーを保持する数に設定されます。ユーザーの保持は、zfs hold コマンドを使用して設定されます。

groupused@*group*
指定されたグループによって消費される、このデータセットの容量。容量は、ls -l で表示される各ファイルのグループにチャージされます。詳細は、userused@*user* プロパティーを参照してください。

非特権ユーザーは、自身のグループの領域使用状況のみを確認できます。root ユーザー、または zfs allow で groupuse 特権を付与されているユーザーは、あらゆるグループの使用状況にアクセスできます。

volblocksize=*blocksize*
ボリュームの場合に、ボリュームのブロックサイズを指定します。ボリュームが書き込まれたあとで、blocksize を変更することはできません。このため、このプロパティーはボリュームの作成時に設定してください。ボリュームのデフォルトの blocksize は 8K バイトです。512 バイト - 1M バイトの範囲で、任意の 2 の累乗を指定できます。

このプロパティーは、列の短縮名 volblock を使用して参照することもできます。

次のネイティブプロパティーを使用すると、ZFS データセットの動作を変更できます。

aclmode=discard | mask | passthrough
chmod(2) の実行時の ACL の変更方法を制御します。「aclmode」プロパティーが「discard」(デフォルト) であるファイルシステムでは、ファイルのモードを表さない ACL エントリがすべて削除されます。「aclmode」プロパティーが「mask」の場合、ユーザーまたはグループアクセス権を削減します。アクセス権は、グループアクセス権ビットと同程度にまで削減されます。ただし、アクセス権がファイルまたはディレクトリの所有者と同じ UID を持つユーザーエントリである場合を除きます。この場合、ACL アクセス権は、所有者のアクセス権ビットと同程度にまで削減されます。mask はモード変更の前後で ACL を保持し、次のモード変更があるまでの間 (chmod(1) によって) ACL を明示的に設定しません。aclmode プロパティーが passthrough であるファイルシステムでは、ファイルまたはディレクトリの新規モードを表す必須の ACL エントリを生成する以外、ACL に変更は加えられません。

aclinherit=discard | noallow | restricted | passthrough | passthrough-x
ファイルとディレクトリが作成されるときに ACL エントリをどのように継承するかを制御します。aclinherit プロパティーが discard であるファイルシステムは、ACL エントリを一切継承しません。aclinherit プロパティーが noallow であ

るファイルシステムは、「deny」アクセス権を指定する、継承可能な ACL エントリのみを継承します。プロパティー値 restricted (デフォルト) は、ACL エントリの継承時に write_acl および write_owner アクセス権を削除します。aclinherit プロパティー値が passthrough であるファイルシステムは、継承時に ACL エントリに加えられた変更を除く、継承可能なすべての ACL エントリを継承します。aclinherit プロパティー値が passthrough-x であるファイルシステムは、passthrough と同じ意味ですが、ファイル作成モードでも実行ビットが必要とされる場合のみすべての ACE が実行権を継承する点が異なります。

プロパティー値が passthrough に設定されている場合は、継承可能な ACE によって決定されるモードでファイルが作成されます。モードに影響を与える継承可能な ACE が存在しない場合は、アプリケーションから要求されたモードに従ってモードが設定されます。

atime=on | off
ファイルを読み取るときにファイルのアクセス時刻を更新するかどうかを制御します。このプロパティーをオフに設定すると、ファイルを読み取るときに書き込みトラフィックが生成されなくなるため、パフォーマンスが大幅に向上する可能性があります。ただし、メールプログラムなどのユーティリティーが予期しない動作をすることがあります。デフォルト値は on です。

canmount=on | off | noauto
このプロパティーが off に設定されている場合、ファイルシステムはマウントできず、zfs mount -a を実行しても無視されます。このプロパティーを off に設定することは、mountpoint プロパティーを none に設定することに似ていますが、継承可能な通常の mountpoint プロパティーをデータセットが引き続き保持する点が異なります。このプロパティーを off に設定することによって、データセットをプロパティー継承のメカニズムとしてのみ使用できるようになります。canmount=off を設定する 1 つの例として、2 つのデータセットが同じ mountpoint を保持するようにできます。これにより、両方のデータセットの子が同じディレクトリ内に表示されますが、継承する特性は異なります。

noauto オプションを設定すると、データセットのマウントおよびマウント解除は明示的に実行することが必要になります。データセットの作成時やインポート時に、データセットが自動的にマウントされることはありません。また、zfs mount -a コマンドでマウントされることや、zfs unmount -a コマンドでマウント解除されることもありません。

このプロパティーは継承されません。

checksum=on | off | fletcher2, | fletcher4 | sha256 | sha256+mac
データの整合性を検証するために使用するチェックサムを制御します。デフォルト値 on では、適切なアルゴリズムが自動的に選択されます。現在、アルゴリズムは fletcher4 ですが、将来のリリースで変更される可能性があります。値が off の場合、ユーザーデータの整合性チェックが無効になります。チェックサムの無効化は、推奨されていない操作です。

このプロパティーを変更した場合、影響を受けるのは新しく書き込まれるデータのみです。

sha256+mac の値は、暗号化が有効な場合にのみ使用できます。暗号化が有効な場合、チェックサムプロパティーは readonly になり、常に sha256+mac に設定されます。

compression=on | off | lzjb | gzip | gzip-*N* | zle
このデータセットで使用される圧縮アルゴリズムを制御します。lzjb 圧縮アルゴリズムは、優れたデータ圧縮と同時に最適なパフォーマンスを提供します。圧縮を on に設定すると、lzjb 圧縮アルゴリズムが使用されます。gzip 圧縮アルゴリズムは、gzip(1) コマンドと同じ圧縮を使用します。gzip のレベルは、値 gzip-*N* を使用すると指定できます。この *N* は、1 (最速) - 9 (最大圧縮率) の整数です。現在、gzip は gzip-6 に相当します (これは gzip(1) のデフォルトでもある)。

このプロパティーは、列の短縮名 compress を使用して参照することもできます。このプロパティーを変更した場合、影響を受けるのは新しく書き込まれるデータのみです。

copies=1 | 2 | 3
このデータセットに関して保存されるデータのコピー数を制御します。これらのコピーは、ミラーリングや RAID-Z など、プールがもたらす冗長性を補うものです。コピーは、可能であれば別のディスクに格納されます。複数のコピーによって使用される容量は、関連するファイルとデータセットにチャージされます。したがって、used プロパティーが変更され、割り当てと予約が減らされます。

このプロパティーを変更した場合、影響を受けるのは新しく書き込まれるデータのみです。したがって、このオプションは、ファイルシステムの作成時に -o copies=*N* オプションを使用して設定してください。

データセットで暗号化が有効な場合、コピーは最大値の 2 に設定されます。

dedup=on | off | verify | sha256[,verify]
データセットに対して複製解除を適用するかどうかを制御します。デフォルト値は off です。複製解除に使用されるデフォルトのチェックサムは sha256 です (変更される可能性があります)。dedup が有効な場合、dedup チェックサムアルゴリズムによって checksum プロパティーが上書きされます。値を verify に設定することは、sha256,verify を指定することと同等です。

プロパティーが verify に設定されている場合、ZFS は、2 つのブロックに同じ署名があるときはいつでも既存のブロックとバイト単位の比較を行なって、内容が同一であることを確認します。

devices=on | off
このファイルシステムでデバイスノードを開くことができるかどうかを制御します。デフォルト値は on です。

exec=on | off
このファイルシステム内部からプロセスを実行可能かどうかを制御します。デフォルト値は on です。

logbias = latency | throughput
このデータセットに対する同期要求を ZFS で最適化する方法を制御します。logbias が latency に設定されている場合、ZFS はプールに別個のログデバイスが存在すればそれを使用して、短い待ち時間で要求を処理します。logbias が throughput に設定されている場合、ZFS はプールの別個のログデバイスを使用しません。その代わりに、ZFS は大域プールのスループットとリソースの使用効率を優先して同時操作を最適化します。デフォルト値は latency です。

mlslabel=*label* | none
mlslabel プロパティーは、Trusted Extensions が有効なシステムでデータセットをゾーンにマウントできるかどうかを指定する機密ラベルです。ラベル付きのデータセットがラベル付きのゾーンに一致する場合、データセットは、そのラベル付きのゾーンからマウントしてアクセスできます。

mlslabel プロパティーが設定されていない場合、デフォルト値は none です。mlslabel プロパティーを none に設定することは、このプロパティーを削除することと同等です。

Trusted Extensions が有効になっていて、適切な特権がある場合にのみ、mlslabel プロパティーを変更できます。このプロパティーを変更する権利は委任できません。ラベルを上位のラベルに変更したり、初期データセットラベルを設定したりするときには、{PRIV_FILE_UPGRADE_SL} 特権が必要です。ラベルを下位のラベルまたはデフォルト (none) に変更するときには、{PRIV_FILE_DOWNGRADE_SL} 特権が必要です。デフォルト以外のラベルへのデータセットの変更は、データセットがマウントされていない場合にのみ実行できます。デフォルトのラベルのデータセットがラベル付きのゾーンにマウントされると、マウント操作によって、mlslabel プロパティーがそのゾーンのラベルに自動的に設定されます。

Trusted Extensions が有効になっていない場合、デフォルトラベル (none) のデータセットだけをマウントできます。

mountpoint=*path* | none | legacy
このファイルシステムで使用されるマウントポイントを制御します。このプロパティーの使用方法の詳細は、「マウントポイント」の節を参照してください。

ファイルシステムの mountpoint プロパティーを変更すると、そのマウントポイントを継承するファイルシステムおよびそのすべての子がマウント解除されます。新しい値が legacy の場合は、マウントが解除されたままになります。それ以外のときは、プロパティーの古い値が legacy または none だった場合、またはプロパティーが変更される前にマウントされていた場合は、自動的に再マウントされます。また、共有されていたすべてのファイルシステムは、共有が解除されてから新しい場所で共有されます。

nbmand=on | off
詳細は、zfs_share(1M) を参照してください。

primarycache=all | none | metadata
一次キャッシュ (ARC) にキャッシュされる内容を制御します。このプロパティーを all に設定すると、ユーザーデータとメタデータの両方がキャッシュされます。このプロパティーを none に設定すると、ユーザーデータもメタデータもキャッシュされません。このプロパティーを metadata に設定すると、メタデータのみがキャッシュされます。デフォルト値は all。

quota=*size* | none
データセットおよびその子孫が消費できる容量を制限します。このプロパティーにより、使用される容量に対して強い制限値が設定されます。これには、子孫の消費する容量すべて (ファイルシステムとスナップショットを含む) が含まれます。割り当て制限がすでに設定されているデータセットの子孫に割り当て制限を設定した場合は、祖先の割り当て制限は上書きされずに、制限が追加されます。

ボリュームには割り当て制限を設定できません。volsize プロパティーが暗黙的な割り当て制限として機能します。

sync=standard | always | disabled
ファイルシステムトランザクションを同期する程度を指定します。このプロパティーは、データセットの作成時に設定することも動的に設定することもでき、ただちに有効になります。このプロパティーは、次の設定のうちの 1 つを持つことができます。

standard
デフォルトのオプション。同期するファイルシステムトランザクションがインテントログに書き込まれてから、書き込まれたすべてのデバイスがフラッシュされ、データが安定した状態になるように (つまり、デバイスコントローラによってキャッシュされないように) します。

always
各ファイルシステムトランザクションが書き込まれ、安定した記憶領域にフラッシュされます。この値はパフォーマンスに非常に不利な状況をもたらしますが、同期するフィルシステムトランザクションのトラブルシューティングには適切な場合があります。

disabled
同期要求は無効になります。ファイルシステムトランザクションは、次の DMU トランザクショングループの確定時にのみ安定した記憶領域に確定されるため、処理が行われるまで数秒かかる場合があります。この設定はパフォーマンスにとって最適です。ただし、データベースや NFS などのアプリケーションの同期トランザクション要求を ZFS が無視するため、これは非常に危険な設定です。さらに、現在アクティブなルートまたは /var ファイルシステムに対してこの設定を有効にした場合、仕様外の動作、アプリケーションデータの損失、およびリプレイ攻撃に対する脆弱性の増大が生じ

る可能性があります。管理者は、これらのリスクを理解した上でこのオプションを使用するようにしてください。

`userquota@`*user*`=`*size*`|none`
指定されたユーザーによって消費される容量を制限します。`userquota` の容量計算では、`refquota` プロパティーと同様に、スナップショットやクローンなど、子孫データセットによって使用される容量は含まれません。ユーザーの容量消費は `userspace@`*user* プロパティーで特定されます。

ユーザーの割り当て制限が適用されるまでに数秒かかることがあります。この遅延のために、ユーザーが自分の割り当て量を超えてから、システムがそのユーザーの割り当て超過を認識することがあります。この場合、システムは追加の書き込みを拒否するようになり、`EDQUOT` エラーメッセージを表示します。詳細は、`zfs userspace` サブコマンドを参照してください。

非特権ユーザーは、自身のグループの領域使用状況のみを確認できます。root ユーザー、または `zfs allow` で `userquota` 特権を付与されているユーザーは、あらゆるユーザーの割り当てを取得および設定できます。

このプロパティーは、ボリューム、バージョン 4 よりも古いファイルシステム、またはバージョン 15 よりも古いプールでは使用できません。`userquota@...` プロパティーは `zfs get all` では表示されません。`@` 記号のあとに、次のいずれかの形式でユーザーの名前を付加する必要があります。

- *POSIX* 名 (例: `joe`)
- *POSIX* 数値 *ID* (例: `789`)
- *SID* 名 (例: `joe.smith@mydomain`)
- *SID* 数値 *ID* (例: `S-1-123-456-789`)

`groupquota@`*group*`=`*size*`|none`
指定されたグループによって消費される容量を制限します。グループの容量消費は `userquota@`*user* プロパティーで特定されます。

非特権ユーザーは、自身のグループの領域使用状況のみを確認できます。root ユーザー、または `zfs allow` で `groupquota` 特権を付与されているユーザーは、あらゆるグループの割り当てを取得および設定できます。

`readonly=on|off`
このデータを変更できるかどうかを制御します。デフォルト値は `off` です。

このプロパティーは、列の短縮名 `rdonly` を使用して参照することもできます。

`recordsize=`*size*
ファイルシステムに格納するファイルの推奨ブロックサイズを指定します。このプロパティーは、レコードサイズが固定されているファイルにアクセスするデータベースワークロードだけで使用するように設計されています。ZFS では、標準的なアクセスパターンに最適化された内部アルゴリズムに従って、ブロックサイズが自動的に調整されます。

作成されるファイルのサイズが非常に大きく、それらのファイルにさまざまなパターンの小さなブロック単位でアクセスするデータベースの場合には、このようなアルゴリズムが最適でないことがあります。recordsize にデータベースのレコードサイズ以上の値を設定すると、パフォーマンスが大きく向上することがあります。このプロパティーを汎用目的のファイルシステムに使用することは、パフォーマンスが低下する可能性があるため、できるだけ避けてください。

デフォルトのレコードサイズは 128K バイトです。指定するサイズは、512 バイト - 1M バイトの 2 の累乗にしてください。

ファイルシステムの recordsize の変更は、その後に作成したファイルにのみ適用され、既存のファイルと受信データには影響しません。

このプロパティーは、列の短縮名 recsize を使用して参照することもできます。

refquota=*size* | none

1 つのデータセットが消費できる容量を制限します。このプロパティーにより、使用される容量に対して強い制限値が設定されます。この強い制限値には、ファイルシステムやスナップショットなどの子孫で使用される容量は含まれません。

refreservation=*size* | none

データセットに保証される最小容量。子孫で使用される容量は含まれません。usedbydataset の容量がこの値を下回っているデータセットは、refreservation で指定された容量を使用していると見なされます。usedbyrefreservation の数値は、この追加の容量を表します。この値はデータセットにチャージされる used の合計容量に加算され、さらに親データセットの使用状況、割り当て、および予約のために消費されます。これにより、将来の書き込み用の容量があらかじめ予約されるようにして、プールリソースの過剰な割り当てからデータセットを保護します。

スナップショットと共有される容量は、あとで新しいデータに置き換えることが可能で、そのスナップショットは、両方のコピーが確実に保持されることを表します。refreservation が設定されている場合、新しいスナップショットを取得するときに usedbyrefreservation を refreservation と同じサイズにまで増やして、コピーが確実に保持されるようにする必要があります。この増加に使用できる十分な容量がデータセットにない場合、スナップショットの作成は拒否されます。

このプロパティーは、列の短縮名 refreserv を使用して参照することもできます。

reservation=*size* | none

データセットおよびその子孫に保証される最小容量。使用している容量がこの値を下回っているデータセットは、予約に指定された容量を使用していると見なされます。予約は、親データセットが使用する容量に計上されるので、親データセットの割り当てと予約を減らすことになります。

このプロパティーは、列の短縮名 reserv を使用して参照することもできます。

rstchown=on | off
ユーザーが chown(1) または chown(2) システムコールでファイルの所有権を与えることをファイルシステムで制限するかどうかを指定します。デフォルトでは chown を制限します。rstchown が off になっている場合、chown はユーザーが PRIV_FILE_CHOWN_SELF 特権を持っている場合と同様に動作します。

secondarycache=all | none | metadata
二次キャッシュ (L2ARC) にキャッシュされる内容を制御します。このプロパティーを all に設定すると、ユーザーデータとメタデータの両方がキャッシュされます。このプロパティーを none に設定すると、ユーザーデータもメタデータもキャッシュされません。このプロパティーを metadata に設定すると、メタデータのみがキャッシュされます。デフォルト値は all。

setuid=on | off
設定された UID ビットが、このシステムで順守されるかどうかを制御します。デフォルト値は on です。

shadow=*URI* | none
ZFS ファイルシステムを、*URI* によって記述されたファイルシステムの「シャドウ」として識別します。 データは、*URI* によって識別されるファイルシステムから、このプロパティーが設定されたシャドウファイルシステムに移行します。移行するファイルシステムは、完全に移行するために読み取り専用にする必要があります。

まだ移行されていないシャドウファイルシステム内のディレクトリへのアクセスは、ディレクトリ全体が移行されるまでブロックされます。まだ移行されていないシャドウファイルシステム内のファイルにアクセスすると、アクセスしたファイルの一部のみが移行されます。複数のプロセスによって、ファイルの異なる部分を同時に移行できます。

2 つの形式の URI を使用できます。1 つは同じ物理システム上でローカルファイルシステムから別のファイルシステムへの移行用で、もう 1 つは NFS サーバーからのリモートでのファイルシステムの移行用です。形式は次のとおりです。

file:///*path*
nfs://*host*/*path*

移行が完了したとき shadowd(1M) がまだ実行中の場合、ファイルシステムは自動的に再マウントされ、シャドウプロパティーは none に設定されます。または、移行が完了したとき、シャドウプロパティーを手動で none に設定できます。

sharenfs=on | off
詳細は、zfs_share(1M) を参照してください。

sharesmb=on | off
詳細は、zfs_share(1M) を参照してください。

snapdir=hidden | visible
: ファイルシステムのルートで、.zfs ディレクトリを非表示にするか、表示するかを制御します。詳細は、「スナップショット」の節を参照してください。デフォルト値は hidden です。

version=1|2|current
: このファイルシステムのディスク上バージョン。プールのバージョンとは無関係です。このプロパティーに設定できるのは、より新しい、サポートされているバージョンのみです。zfs upgrade コマンドを参照してください。

volsize=*size*
: ボリュームの論理サイズを指定します。デフォルトでは、ボリュームを作成するときに確立される refreservation は、ZFS メタデータのオーバーヘッドを考慮に入れるため、実際の論理ボリュームサイズよりもいくらか大きくなります。volsize への変更があった場合には、refreservation にも同様の変更が反映されます。volsize に設定可能な値は、volblocksize の倍数だけです。この値をゼロにすることはできません。

  refreservation は、利用者の予期しない動作を避けるためにボリュームに設定されます。予約を使用しない場合、ボリュームの使用方法によっては、ボリュームの容量が不足して未定義の動作またはデータ破壊が発生する可能性があります。このような影響は、ボリュームの使用中にボリュームサイズを変更した場合にも発生することがあります。特に、サイズを縮小した場合にはその可能性が高くなります。 ボリュームサイズを調整するときは、特に注意するようにしてください。

  推奨されませんが、zfs create -V に -s オプションを指定して「疎ボリューム」(シンプロビジョニングとしても知られる) を作成できます。「疎ボリューム」とは、予約がボリュームサイズよりも小さいボリュームのことです。このため、領域上のプールが小さい場合、疎ボリュームへの書き込みが ENOSPC で失敗する可能性があります。疎ボリュームの場合、volsize を変更しても予約には反映されません。

vscan=on|off
: 詳細は、zfs_share(1M) を参照してください。

xattr=on | off
: このファイルシステムで拡張属性が有効かどうかを制御します。デフォルト値は on です。

zoned=on | off
: データセットを非大域ゾーンから管理するかどうかを制御します。詳細は、「ゾーン」の節を参照してください。デフォルト値は off です。

次のプロパティーは、ファイルシステムの作成後には変更できないため、ファイルシステムの作成時に設定するようにしてください。zfs create または zpool create コマンドで設定されていない場合、これらのプロパティーは親データセットから継承されます。これらの機能がサポートされる前に親データセットが作成されたた

め、親データセットにこれらのプロパティーが存在しない場合、新しいファイルシステムではこれらのプロパティーにデフォルト値が設定されます。

`casesensitivity = sensitive | insensitive | mixed`
詳細は、zfs_share(1M) を参照してください。

`normalization = none | formC | formD | formKC | formKD`
詳細は、zfs_share(1M) を参照してください。

`utf8only =on | off`
詳細は、zfs_share(1M) を参照してください。

`encryption=off | on | aes-128-ccm | aes-192-ccm | aes-256-ccm | aes-128-gcm | aes-192-gcm | aes-256-gcm`
詳細は、zfs_encrypt(1M) を参照してください。

次のプロパティーは、作成時に指定する必要があり、特殊コマンドを使用すると変更できます。

`keysource=`*raw* | *hex* | *passphrase,prompt* | *file*
詳細は、zfs_encrypt(1M) を参照してください。

一時的なマウントポイントプロパティー

ファイルシステムのマウント時に、従来の mount(1M) コマンドまたは zfs mount コマンドを使用して、プロパティーに応じたマウントオプションが設定されます。プロパティーとマウントオプションは、次のような関係になっています。

```
PROPERTY                MOUNT OPTION
 devices                 devices/nodevices
 exec                    exec/noexec
 readonly                ro/rw
 setuid                  setuid/nosetuid
 xattr                   xattr/noxattr
 rstchown                rstchown/norstchown
```

さらに、-o オプションを使って、ディスクに格納されたプロパティーに影響を及ぼすことなく、これらのオプションをマウントごとに設定できます。コマンド行に指定した値により、データセットに格納された値が上書きされます。-nosuid オプションは、nodevices,nosetuid の別名です。これらのプロパティーは、zfs get コマンドにより、「temporary」と報告されます。mountpoint 以外のプロパティーの場合、データセットのマウント時にプロパティーが変更されると、新規設定により一時設定がすべて上書きされます。mountpoint プロパティーは、一時的な mountpoint プロパティーが有効である間 (つまり、データセットが一時的な場所にマウントされている間) は変更できません。

ユーザープロパティー

ZFS は、標準のネイティブプロパティーに加えて、任意のユーザープロパティーもサポートします。ユーザープロパティーは ZFS の動作には影響を与えませんが、アプリケーションや管理者が使用して、データセット (ファイルシステム、ボリューム、およびスナップショット) に注釈を付けることができます。

ユーザープロパティー名には、ネイティブプロパティーと区別するためにコロン (:) 文字を含める必要があります。ユーザープロパティーに含めることができるのは、小文字の英字、数字、および句読文字のコロン (:)、ダッシュ (-)、ピリオド (.)、および下線 (_) です。想定されている規則では、プロパティー名は 2 つの部分に分割します (例: *module*: *property*)。ただし、この名前空間は ZFS によって強制されているものではありません。ユーザープロパティー名には、最大 256 文字を使用できます。名前の先頭をダッシュ (-) で始めることはできません。

ユーザープロパティーをプログラムで使用する場合、プロパティー名の *module* コンポーネントには、逆順にした DNS ドメイン名を使用することを強くお勧めします。これは、それぞれ単独で開発された 2 つのパッケージが、異なる目的で同じプロパティー名を使用する可能性を減らすためです。Oracle Solaris リリースでは、`com.oracle` ユーザープロパティーが `beadm` コマンドおよびライブラリ用に予約されています。

ユーザープロパティーの値は任意の文字列であり、常に継承されます。また、決して検証されることがありません。プロパティーを処理するコマンド (`zfs list`、`zfs get`、`zfs set` など) はすべて、ネイティブプロパティーとユーザープロパティーの両方の操作に使用できます。ユーザープロパティーをクリアするには、`zfs inherit` コマンドを使用します。プロパティーがどの親のデータセットにも定義されていない場合は、完全に削除されます。プロパティー値は、1024 文字以内に制限されています。

スワップまたはダンプデバイスとしての ZFS ボリュームの使用

初期インストール中に、ZFS ルートプールの ZFS ボリューム上にスワップデバイスおよびダンプデバイスが作成されます。スワップ領域とダンプデバイスには別個の ZFS ボリュームを使用する必要があります。ZFS ファイルシステム上のファイルには、スワップを作成しないでください。ZFS のスワップファイル設定は、サポートされていません。

スワップデバイスとして使用される ZFS ボリュームは、そのデバイスの `encryption` プロパティーを指定して、`vfstab(4)` の `encrypted` オプションを指定することによって暗号化できます。暗号化プロパティーの詳細は、zfs_encrypt.1m を参照してください。

システムのインストール後またはアップグレード後にスワップ領域やダンプデバイスを変更する必要がある場合は、`swap(1M)` コマンドと `dumpadm(1M)` コマンドを使用します。スワップ領域またはダンプデバイスのサイズを変更する必要がある場合は、『『Oracle Solaris の管理: ZFS ファイルシステム』』を参照してください。

サブコマンド

状態を変更するサブコマンドはすべて、元の形式でプールに永続的に記録されます。

`zfs ?`
ヘルプメッセージを表示します。

`zfs help` *command* | help | *property property-name* | *permission*

`zfs` コマンドの使用法情報を表示します。特定のコマンド、プロパティー、または委任アクセス権のヘルプを表示できます。特定のコマンドまたはプロパティーのヘルプを表示した場合、コマンド構文またはプロパティー値が表示されます。引数を何も付けずに `zfs help` を使用すると、`zfs` コマンドの完全なリストが表示されます。

`zfs create` [`-p`] [`-o` *property=value*] ... *filesystem*

新しい ZFS ファイルシステムを作成します。ファイルシステムは、親から継承した `mountpoint` プロパティーに従って自動的にマウントされます。

`-p`

存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した `mountpoint` プロパティーに従って自動的にマウントされます。コマンド行で `-o` オプションを使用して指定したプロパティーはすべて無視されます。ターゲットのファイルシステムがすでに存在する場合、処理は正常に完了します。

`-o` *property=value*

データセットの作成時にコマンド `zfs set` *property=value* が呼び出された場合と同様に、指定されたプロパティーを設定します。編集可能なすべての ZFS プロパティーは、作成時にも設定可能です。複数の `-o` オプションを指定できます。複数の `-o` オプション内で同じプロパティーを指定した場合は、エラーが発生します。

`zfs create` [`-ps`] [`-b` *blocksize*] [`-o` *property=value*] ... `-V` *size volume*

指定したサイズのボリュームを作成します。ボリュームは、`/dev/zvol/{dsk,rdsk}/`*path* 内にブロックデバイスとしてエクスポートされます。ここで、*path* は ZFS 名前空間内のボリュームの名前です。このサイズは、デバイスによりエクスポートされる論理サイズを表します。デフォルトでは、同サイズの予約が作成されます。

size は、ボリュームが *blocksize* に関係なく整数のブロックを持つように、もっとも近い 128 K バイトに自動的に切り上げられます。

`-p`

存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した `mountpoint` プロパティーに従って自動的にマウントされます。コマンド行で `-o` オプションを使用して指定したプロパティーはすべて無視されます。ターゲットのファイルシステムがすでに存在する場合、処理は正常に完了します。

`-s`

予約なしで疎ボリュームを作成します。疎ボリュームの詳細は、「ネイティブプロパティー」の節の `volsize` を参照してください。

-o *property=value*
データセットの作成時に zfs set *property=value* コマンドが呼び出された場合と同様に、指定されたプロパティーを設定します。編集可能なすべての ZFS プロパティーは、作成時にも設定可能です。複数の -o オプションを指定できます。複数の -o オプション内で同じプロパティーを指定した場合は、エラーが発生します。

-b *blocksize*
-o volblocksize=*blocksize* と同等です。このオプションを -o volblocksize と組み合わせて指定した場合の動作は、定義されていません。

zfs destroy [-rRf] *filesystem*|*volume*
指定されたデータセットを破棄します。デフォルトでは、このコマンドは共有中のファイルシステムすべての共有を解除し、マウント中のファイルシステムすべてをマウント解除し、アクティブな依存関係 (子またはクローン) を持つデータセットの破棄を拒否します。

-r
すべての子を再帰的に破棄します。

-R
ターゲット階層外にあるクローンファイルシステムを含む、すべての依存関係を再帰的に破棄します。

-f
unmount -f コマンドを使用してすべてのファイルシステムを強制的にマウント解除します。非ファイルシステムやマウント解除されたファイルシステムは、このオプションの影響を受けません。

-r や -f オプションを適用すると、プールのかなりの部分を破棄することが可能で、使用中のマウントされたファイルシステムで予期しない動作が発生する場合があるため、これらのオプションは特に注意深く使用するようにしてください。

zfs destroy [-rRd] *snapshot*
指定されたスナップショットが -d オプションを付けない zfs destroy コマンドで破棄される場合にかぎり、そのスナップショットがすぐに破棄されます。このように破棄がただちに実行されるのは、たとえば、スナップショットにクローンがない場合や、ユーザーによる参照数がゼロだった場合です。

すぐに破棄される条件を満たしていないスナップショットには、削除が遅延されるようにマークが付けられます。この状態では、そのスナップショットは使用および表示が可能なスナップショットとして存在し、前述の両方の前提条件が満たされると、その時点で破棄されます。

-d
スナップショットの削除を延期します。

-r
子孫ファイルシステムでこの名前を持つすべてのスナップショットを破棄します (または削除が遅延されるようにマークが付けられます)。

-R
すべての依存関係を再帰的に破棄します。

zfs snapshot [-r] [-o *property=value*] ... *filesystem@snapname*|*volume@snapname*
指定した名前のスナップショットを作成します。正常終了したシステムコールによってファイルシステムに加えられていた変更は、すべてスナップショットに含まれます。zfs snapshot の別名として、zfs snap を使用できます。詳細は、「スナップショット」の節を参照してください。

-r
すべての子孫データセットのスナップショットを再帰的に作成します。スナップショットは原子的に取得されるため、再帰的スナップショットはすべて同じ時点のものになります。

-o *property=value*
指定されたプロパティーを設定します。詳細は、zfs create を参照してください。

zfs rollback [-rRf] *snapshot*
指定したデータセットを以前のスナップショットにロールバックします。データセットをロールバックすると、スナップショット作成時から変更されたすべてのデータは破棄され、データセットがスナップショット作成時の状態に戻ります。デフォルトでは、このコマンドを使って、最新のスナップショット以外のスナップショットにロールバックすることはできません。最新でないスナップショットにロールバックする場合は、-r オプションを使って中間スナップショットをすべて破棄する必要があります。

-rR オプションは、再帰的スナップショットの子スナップショットを再帰的に破棄しません。これらのオプションはどちらも、最上位の再帰的スナップショットだけを破棄します。再帰的スナップショットを完全にロールバックするには、個々の子スナップショットをロールバックする必要があります。

-r
指定したスナップショット以降のスナップショットをすべて再帰的に破棄します。

-R
指定したスナップショット以降のスナップショットを再帰的に破棄すると共に、これらのスナップショットのクローンもすべて破棄します。

-f
-R オプションとともに使用すると、破棄するクローンファイルシステムをすべて強制的にマウント解除します。

`zfs clone` [`-p`] [`-K`] [`-o` *property=value*] ... *snapshot filesystem*|*volume*

指定したスナップショットのクローンを作成します。詳細は、「クローン」の節を参照してください。ターゲットのデータセットは、ZFS 階層内の任意の場所に配置できます。作成されたデータセットは元のデータセットと同タイプになります。

`-p`

存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した `mountpoint` プロパティーに従って自動的にマウントされます。ターゲットのファイルシステムまたはボリュームがすでに存在する場合、処理は正常に完了します。

`-o` *property=value*

指定されたプロパティーを設定します。詳細は、`zfs create` を参照してください。

`-K`

詳細は、`zfs_encrypt(1M)` を参照してください。

`zfs promote` *clone-filesystem*

特定のクローンファイルシステムへの移行を促し、そのファイルシステムが「元」のスナップショットに依存しないようにします。これにより、そのクローンの作成元のファイルシステムを破棄できるようになります。クローンの親子依存関係が逆転し、元のファイルシステムが、指定されたファイルシステムのクローンになります。

クローンされたスナップショットと、そのスナップショットより前のすべてのスナップショットは、移行を促されたクローンによって所有されるようになります。それらが使用する領域は、元のファイルシステムから移行を促されたクローンに移されます。したがって、それらのスナップショットを収容するための十分な領域が必要です。この操作を行なっても新しい領域は消費されませんが、領域のアカウンティングは調整されます。移行を促されたクローンは、自身の相反するスナップショット名を持ってはいけません。`rename` サブコマンドを使えば、競合するスナップショット名を変更できます。

`zfs rename` *filesystem*|*volume*|*snapshot*
filesystem|*volume*|*snapshot*
`zfs rename` [`-p`] *filesystem*|*volume filesystem*|*volume*

指定したデータセットの名前を変更します。スナップショットを除き、新規ターゲットは ZFS 階層内の任意の場所に配置できます。スナップショットの名前を変更できるのは、親のファイルシステムまたはボリューム内だけです。スナップショットの名前を変更する場合、スナップショットの親ファイルシステムを 2 番目の引数として指定する必要はありません。名前の変更されたファイルシステムは、新しいマウントポイントを継承できます。この場合、ファイルシステムは、マウント解除されてから新しいマウントポイントで再マウントされます。

-p
存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した mountpoint プロパティーに従って自動的にマウントされます。

zfs rename -r *snapshot snapshot*
すべての子孫データセットのスナップショットの名前を再帰的に変更します。スナップショットは、再帰的に名前を変更できるデータセットにすぎません。

zfs list [-rH|-d *depth*] [-o *property*[,...]] [-t *type*[, ...]] [-s *property*] ... [-S *property*] ... [*filesystem*|*volume*|*snapshot*|*path*] ...
指定したデータセットのプロパティー情報を、表形式で表示します。指定した場合、プロパティー情報を絶対パス名または相対パス名で表示できます。デフォルトでは、すべてのファイルシステムとボリュームが表示されます。スナップショットは、listsnaps プロパティーが on の場合に表示されます (デフォルトは off)。name、used、available、referenced、および mountpoint の各フィールドが表示されます。

-H
スクリプティングモードで使用します。ヘッダーを出力せず、フィールドを任意のスペース文字ではなく単一のタブで区切ります。

-r
データセットのすべての子をコマンド行に再帰的に表示します。

-d *depth*
データセットのすべての子を再帰的に表示します。再帰は最大 *depth* に制限されます。深さ 1 の場合は、データセットとその直接の子だけが表示されます。

-o *property*
表示するプロパティーのコンマ区切りのリスト。次のプロパティーを指定する必要があります。

- 「ネイティブプロパティー」の節で示したプロパティーのいずれか
- ユーザープロパティー
- データセット名を表示する値 name
- ファイルシステムとボリュームの領域使用状況プロパティーを表示する値 space。これは、-o name,avail,used,usedsnap,usedds,usedrefreserv,usedchild-t filesystem,
構文のショートカットです。

-s *property*
プロパティーの値に基づいて、出力を列で昇順にソートするためのプロパティー。プロパティーは、「プロパティー」の節で示したプロパティーのいずれか、またはデータセット名をソートする特殊な値 name にする必要があり

ます。複数の -s プロパティーオプションを使用することで、一度に複数のプロパティーを指定できます。複数の -s オプションは、左から右の順で、重要度の高さが判断されます。

次に、ソート基準の一覧を示します。

- 数値型は、数値順にソートされます。
- 文字列型は、アルファベット順にソートされます。
- 行に適さない型は、指定したソート順に関係なく、その行がリテラルソート順のいちばん下になります。
- ソートオプションを指定しない場合は、zfs list の既存の動作が維持されます。

-S *property*
: -s オプションと同じですが、プロパティーで降順にソートされる点が異なります。

-t *type*
: 表示するタイプのコンマ区切りのリスト。ここで、*type* は、filesystem、snapshot、volume、または all のいずれかになります。たとえば、-t snapshot を指定すると、スナップショットのみが表示されます。タイプ指定子の代わりに、fs (ファイルシステム)、snap (スナップショット)、および vol (ボリューム) の各別名を使用できます。

zfs set [-r] *property=value filesystem|volume|snapshot* ...
: プロパティーを、各データセット用に指定した値に設定します。一部のプロパティーのみ、編集可能です。設定可能なプロパティーおよび有効な値については、「プロパティー」の節を参照してください。数値は、正確な値を指定することも、B、K、M、G、T、P、E、または Z (それぞれ、バイト、K バイト、M バイト、G バイト、T バイト、P バイト、E バイト、または Z バイトを表す) のサフィックスを使って人間の読み取り可能な形式で指定することもできます。スナップショットでユーザープロパティーを設定できます。詳細は、「ユーザープロパティー」の節を参照してください。

-r
: 子データセットのサブツリー全体に対して、設定の有効な値を再帰的に適用します。有効な値は、プロパティーに応じて設定または継承できます。

zfs get [-rHp|-d *depth*] [-o all | *field*[,...] [-s *source*[,...]] all | *property*[,...] *filesystem|volume|snapshot* ...
: 指定したデータセットのプロパティーを表示します。データセットが指定されていない場合、このコマンドはシステムのすべてのデータセットのプロパティーを表示します。プロパティーごとに、次の列が表示されます。

```
name      Dataset name
property  Property name
value     Property value
source    Property source. Can either be local, default,
```

```
temporary, inherited, or none (-).
```

デフォルトでは、RECEIVED 列を除くすべての列が表示されます。特定の列またはすべての列を指定するには、-o オプションを使用します。このコマンドには、「ネイティブプロパティー」および「ユーザープロパティー」で説明されているコンマ区切りのプロパティーリストを指定できます。

特殊な値 all を使って、指定したデータセットのタイプ (ファイルシステム、ボリューム、またはスナップショット) に適用されるプロパティーをすべて表示できます。

-r
任意の子のプロパティーを再帰的に表示します。

-d *depth*
データセットのすべての子を再帰的に表示しますが、再帰の深さは *depth* に制限されます。深さ 1 の場合は、データセットとその直接の子だけが表示されます。

-H
スクリプトによる解析がより容易な形式で、出力を表示します。ヘッダーがすべて省略され、フィールドが任意の数のスペースではなく、タブ 1 つで明示的に区切られます。

-o *field*
表示するフィールドのセット。次の 1 つ以上のフィールドを指定します。

```
name,property,value,received,source
```

複数のフィールドをコンマ区切りリストで表します。デフォルト値は次のとおりです。

```
name,property,value,source
```

キーワード all はすべてのソースを指定します。

-s *source*
表示するソースのコンマ区切りのリスト。このリストにないソースからのプロパティーは、無視されます。各ソースは、

```
local,default,inherited,temporary,received,none
```

デフォルト値はすべてのソースです。

-p
解析可能な (絶対) 値で数を表示します。

zfs set share=*name*=*sharename*,path=*pathname*,[desc=*description*],prot=smb | nfs[,*property*=*value*[,...]] *filesystem*
zfs set share の構文についての完全な説明は、zfs_share(1M) を参照してください。

zfs inherit [-rS] *property filesystem*|*volume*|*snapshot* ...

指定したプロパティーをクリアして、そのプロパティーが祖先から継承されるようにします。祖先にプロパティーが設定されていない場合は、デフォルト値が使用されます。デフォルト値のリスト、および継承可能なプロパティーの詳細については、「プロパティー」の節を参照してください。

-r

すべての子で指定したプロパティーを再帰的に継承します。

-S

受信したプロパティー値があれば、その値に戻します。プロパティーに受信値がない場合、zfs inherit -S の動作は、-S を指定しない zfs inherit と同じです。プロパティーに受信値が存在する場合、zfs inherit は、zfs inherit -S によって継承値が受信値に戻されるまで、受信値を継承値でマスクします。

zfs upgrade

ファイルシステムバージョンを指定します。このバージョンによって、現在実行中のソフトウェアリリースで使用可能なファイルシステム機能が決まります。古いファイルシステムバージョンを使用し続けることはできますが、一部の機能が使用できなくなる可能性があります。ファイルシステムは、zfs upgrade - a コマンドを使用するとアップグレードできます。以前のソフトウェアバージョンを実行しているシステムで、以降のバージョンのファイルシステムにアクセスすることはできません。

zfs upgrade [-v]

現在のソフトウェアでサポートされている ZFS ファイルシステムのバージョンを表示します。現在の ZFS ファイルシステムのバージョンおよびサポートされる以前のバージョンすべてが、各バージョンで提供される機能に関する説明とともに表示されます。

zfs upgrade [-r] [-V *version*] [-a | *filesystem*]

ファイルシステムを新しいディスク上バージョンにアップグレードします。ファイルシステムをアップグレードすると、古いソフトウェアバージョンを実行しているシステムではアクセスできなくなります。新しいファイルシステムのスナップショットから生成された zfs send ストリームは、古いソフトウェアバージョンを実行しているシステムではアクセスできません。

一般に、ファイルシステムのバージョンはプールのバージョンとは無関係です。zpool upgrade コマンドの詳細は、zpool(1M) を参照してください。

場合によっては、ファイルシステムのバージョンとプールのバージョンが相互に関連しており、ファイルシステムのバージョンをアップグレードする前にプールのバージョンをアップグレードする必要があります。

-a

インポートされたすべてのプールのファイルシステムをすべてアップグレードします。

filesystem
指定されたファイルシステムをアップグレードします。

-r
指定されたファイルシステムおよびすべての子孫ファイルシステムをアップグレードします。

-V *version*
指定された *version* にアップグレードします。-V フラグが指定されていない場合、このコマンドは最新のバージョンにアップグレードします。このオプションは、バージョン番号を増やす場合にのみ使用できます。また、指定できるのは、このソフトウェアでサポートされている最新のバージョンまでです。

zfs userspace [-hniHp] [-o *field*[,...]] [-sS *field*]... [-t *type* [,...]] *filesystem* | *snapshot*
指定されたファイルシステムまたはスナップショットでの、各ユーザーの消費容量および割り当てを表示します。これは、userused@*user* プロパティーと userquota@*user* プロパティーに対応します。

-h
構文のヘルプメッセージを表示して終了します。

-n
ユーザー名やグループ名の代わりに数値 ID を出力します。

-H
ヘッダーを出力せず、タブ区切りの出力を使用します。

-p
正確な (解析可能な) 数値出力を使用します。

-o *field*[,...]
次のセットから、指定されたフィールドのみを表示します。type,name,used,quota。デフォルトでは、すべてのフィールドが表示されます。

-s *field*
出力をこのフィールドでソートします。*s* フラグと *S* フラグは複数回指定できます。その場合、まず 1 つのフィールドでソートしてから、順次別のフィールドでソートできます。デフォルトは -s type -s name です。

-S *field*
このフィールドで逆順にソートします。-s を参照してください。

-t *type*[,...]
次のセットから、指定されたタイプのみを出力します。all,posixuser,smbuser,posixgroup,smbgroup.

デフォルトは -t posixuser,smbuser です。

デフォルトを変更してグループタイプを含めることもできます。

-i
SID を POSIX ID に変換します。マッピングが存在しない場合、POSIX ID は一時的なものになることがあります。通常の POSIX インタフェース (stat(2)、ls -l など) はこの変換を実行するため、-i オプションを使用すると、zfs userspace からの出力をこれらのユーティリティーと直接比較できます。ただし、SMB から POSIX への名前のマッピングが確立される前に SMB ユーザーによって作成されたファイルがある場合は、-i を使用すると混乱が生じることがあります。そのような場合は、SMB エンティティーが所有するファイルと、POSIX エンティティーが所有するファイルが混在します。しかし、-i オプションでは、どちらについてもすべて POSIX エンティティーの使用量および割り当てとして報告されます。

zfs groupspace [-hniHp] [-o *field*[,...]] [-sS *field*]... [-t *type* [,...]] *filesystem* | *snapshot*
指定されたファイルシステムまたはスナップショットでの、各グループの消費容量および割り当てを表示します。このサブコマンドは zfs userspace と同じですが、デフォルトで表示されるタイプは -t posixgroup,smbgroup です。

-

zfs mount
現在マウントされているすべての ZFS ファイルシステムを表示します。

zfs mount [-vOS] [-o *options*] -a | *filesystem*
ZFS ファイルシステムをマウントします。これは、ブートプロセスの一部として自動的に呼び出されます。

-o *options*
マウント時に一時的に使用する、コンマ区切りのマウントオプションリスト (省略可能)。詳細は、「一時的なマウントポイントプロパティー」の節を参照してください。

-O
オーバーレイマウントを実行します。詳細は、mount(1M) を参照してください。

-v
マウントの進行状況を報告します。

-a
使用可能なすべての ZFS ファイルシステムをマウントします。これは、ブートプロセスの一部として自動的に呼び出されます。

filesystem
指定されたファイルシステムをマウントします。

暗号化されたデータセットの zfs mount 操作では、keysource プロパティー値によっては鍵の入力が求められる場合があります。これは、たとえば keysource ロケータが prompt に設定されている場合に起こることがあります。

`zfs unmount [-f] -a | filesystem|mountpoint`

現在マウントされている ZFS ファイルシステムをマウント解除します。これは、シャットダウンプロセスの一部として自動的に呼び出されます。

`-f`

ファイルシステムを、使用中であっても強制的にマウント解除します。

`-a`

使用可能なすべての ZFS ファイルシステムをマウント解除します。これは、ブートプロセスの一部として自動的に呼び出されます。

filesystem|mountpoint

指定されたファイルシステムをマウント解除します。このコマンドには、システム上の ZFS ファイルシステムのマウントポイントのパスを指定することもできます。

暗号化されたデータセットの場合、ファイルシステムがマウント解除されているときには、鍵の読み込みは解除されません。鍵を読み込み解除するには、`zfs key` を参照してください。

`zfs share -a | dataset`

`zfs share` の構文および例についての完全な説明は、`zfs_share(1M)` を参照してください。

`zfs unshare -a | dataset|mountpoint`

`zfs unshare` の構文および例についての完全な説明は、`zfs_share(1M)` を参照してください。

`zfs send [-DvRbp] [-[iI] snapshot] snapshot`

`zfs send -r [-Dvbcp] [-i snapshot] snapshot`

2 番目の *snapshot* のストリーム表現を作成します。これは、標準出力に書き込まれます。出力は、`ssh(1)` などを使用して、ファイルまたは別のシステムにリダイレクトできます。デフォルトでは、完全なストリームが生成されます。

`-b`

データセットが受信された場合にのみ、ローカル設定によって上書きされているかどうかに関係なく、受信したプロパティー値だけを送信します。`zfs receive` で、送信されたデータセット上でバックアップされた受信プロパティーを復元し、かつソースデータセットそのものではなくデータのバックアップ方法にのみ関係する可能性があるローカル設定を送信しないようにする場合に、このオプションを使用します。

`-c`

自己完結型ストリームを作成します。自己完結型ストリームとは、ストリームパッケージに含まれていないデータセットに依存しないストリームのことです。`-r` オプションでのみ有効です。`-i` オプションで使用した場合、ストリームは、そのオプションの引数として指定したスナップショットに依存します。

詳細は、『ZFS 管理ガイド』の ZFS ストリームに関する節を参照してください。

`-D`
ストリームで dedup 処理を実行します。複製解除されたストリームは、ストリームの複製解除機能をサポートしていないシステム上では受信できません。

複製ストリームパッケージと再帰的ストリームパッケージの違いについて理解するには『ZFS 管理ガイド』の ZFS ストリームに関する節を参照してください。

`-i` *snapshot*
最初の *snapshot* から 2 番目の *snapshot* への増分ストリームを生成します。増分ソース (最初の *snapshot*) は、スナップショット名の最後のコンポーネント (たとえば、`@` のあとの部分) として指定できます。これは、2 番目の *snapshot* と同じファイルシステムに由来すると見なされます。

宛先がクローンの場合、ソースとして元のスナップショットを指定できますが、完全な形式で指定する必要があります (たとえば、単に `@origin` と指定するのではなく、`pool/fs@origin` とする)。

`-I` *snapshot*
最初のスナップショットから 2 番目のスナップショットにすべての中間スナップショットを送信するストリームパッケージを生成します。たとえば、`-I @a fs@d` は `-i @a fs@b; -i @b fs@c; -i @c fs@d` に似ています。増分ソーススナップショットは、`-i` オプションと同様に指定できます。

`-R`
複製ストリームパッケージを生成します。これにより、指定されたファイルシステム、および指定されたスナップショットまでのすべての子孫ファイルシステムが複製されます。受信時には、すべてのプロパティー、スナップショット、下位ファイルシステム、およびクローンが維持されます。

`-i` または `-I` フラグを `-R` フラグと組み合わせて使用すると、増分複製ストリームが生成されます。プロパティーの現在の値、および現在のスナップショット名とファイルシステム名は、ストリームの受信時に設定されます。このストリームの受信時に `-F` フラグを指定すると、送信側に存在しないスナップショットとファイルシステムは破棄されます。

`-r`
再帰的ストリームパッケージを生成します。再帰的ストリームパッケージには、一連の完全ストリームまたは増分ストリームあるいはその両方が含まれています。受信時に、すべてのプロパティーおよび子孫ファイルシステムが保持されます。`-R` フラグで生成される複製ストリームパッケージとは異なり、中間スナップショットは、それがストリームに含まれる複製元でないかぎり保持されません。

-i オプションと -r オプションを組み合わせて使用すると、再帰的増分ストリームが生成されます。プロパティーの現在の値、および現在のスナップショット名とファイルシステム名は、ストリームの受信時に設定されます。このストリームの受信時に -F オプションを指定すると、送信側に存在しないスナップショットとファイルシステムは破棄されます。-I オプションは -r オプションと組み合わせて使用できません。

-c オプションと組み合わせると、自己完結型の再帰的ストリームパッケージが作成されます。-c と -i の両方のオプションを使用した場合、-i オプションで指定したスナップショットを持たないファイルシステムおよびボリュームは、自己完結型ストリームとして送信されます。

再帰的ストリームパッケージと複製ストリームパッケージの違いについて理解するには『ZFS 管理ガイド』の ZFS ストリームに関する節を参照してください。

-p
プロパティーを送信します。

-v
生成されたストリームパッケージに関する詳細な情報を出力します。

ストリームの形式が確定されました。ZFS の将来のバージョンで、使用しているストリームを受信できます。

zfs receive [-vnFu] [[-o *property=value*] | [-x *property*]] ... *filesystem*|*volume*| *snapshot*
zfs receive [-vnFu] [[-o *property=value*] | [-x *property*]] ... [-d | -e] *filesystem*

スナップショットを作成します。内容は、標準入力から受信したストリームの指定に基づきます。ストリーム全体を受信する場合、新しいファイルシステムも作成されます。ストリームは zfs send サブコマンドによって作成されます。このサブコマンドはデフォルトで、完全なストリームを作成します。zfs receive の別名として、zfs recv を使用できます。

増分ストリームが受信された場合、対象となるファイルシステムがすでに存在しており、かつその最新のスナップショットがその増分ストリームのソースに一致する必要があります。ZFS ボリュームの場合、宛先のデバイスリンクが破棄されて再作成されます。つまり、receive 操作中はボリュームにアクセスできません。

zfs send -R コマンドによって生成されたスナップショット複製パッケージストリームが受信される場合、送信場所に存在していないスナップショットは、zfs destroy -d コマンドを使用することによって破棄されます。-o *property=value* または -x プロパティーが指定された場合、その指定は、複製されたデータセットのサブツリー全体にわたって、プロパティーの有効な値に適用されます。有効なプロパティー値は、そのプロパティー、および複製されたサブツリーのいちばん上にデータセットがあるかどうかに応じて、設定または継承できます。受信されたプロパティーは上書きされるものの保持され、zfs inherit -rS または zfs send -Rb で復元できます。

このサブコマンドが作成するスナップショット (ストリーム全体を受信する場合はファイルシステムも) の名前は、引数タイプと -d または -e オプションによって決まります。

引数がスナップショット名の場合には、指定した *snapshot* が作成されます。引数がファイルシステム名またはボリューム名の場合には、送信されたスナップショットと同名のスナップショットが、指定した *filesystem* または *volume* 内部で作成されます。-d または -e オプションを指定すると、指定した filesystem に送信したスナップショットの名前を追加してスナップショット名が決定されます。-d オプションが指定された場合、送信されたスナップショットパスのプール名を除くすべてが付加され (たとえば、送信されたスナップショット a/b/c@1 の b/c@1 が付加)、-e オプションが指定された場合は、送信されたスナップショットパスの末尾だけが付加されます (たとえば、送信されたスナップショット a/b/c@1 の c@1 が付加)。-d の場合、送信されたスナップショットのパスを複製するために必要なファイルシステムは、指定されたファイルシステム内で作成されます。

-d
: 前の段落で説明したように、送信されたスナップショットパスの最初の要素を除くすべて (プール名を除くすべて) を使用して、新しいスナップショットの名前が決定されます。

-e
: 前の段落で説明したように、送信したスナップショットパスの最後の要素を使用して、新しいスナップショットの名前が決定されます。

-F
: 受信操作を実行する前に、ファイルシステムを最新のスナップショットに強制的にロールバックします。増分複製ストリーム (zfs send -R -[iI] で生成されたものなど) を受信する場合は、送信側に存在しないスナップショットとファイルシステムを破棄します。

-n
: ストリームを実際には受信しません。これを -v オプションと組み合わせて使用すると、受信操作で使用する名前を確認するのに役に立ちます。

-o *property=value*
: 非増分 send ストリームからの受信データセットの作成時または増分 send ストリームからの受信データセットの更新時にコマンド zfs set *property=value* が呼び出された場合と同様に、指定されたプロパティーを設定します。編集可能なすべての ZFS プロパティーは、受信時にも設定可能です。normalization や casesensitivity など受信データにバインドされるプロパティーは一度だけ設定可能で、データセットが zfs receive によって新しく作成された場合でも、受信時には設定できません。複数の -o オプションを指定できます。複数の -o オプションまたは -x オプション内で同じプロパティーを指定した場合は、エラーが発生します。

-u
受信したストリームに関連付けられているファイルシステムをマウントしません。

-v
ストリームおよび受信操作の所要時間に関する詳細な情報を出力します。

-x *property*
receive のあとにある指定されたプロパティーの有効な値が、そのプロパティーが send ストリームから除外されている場合と同様に、send ストリーム内のそのプロパティー (もしあれば) の値に影響されないようにします。指定されたプロパティーが send ストリームに存在しない場合、このオプションは何も行いません。受信されたプロパティーを上書きする必要がある場合、有効な値をそのプロパティーに応じて設定または継承できます。増分更新の場合、-x は、既存のローカル設定または明示的な継承を変更せずに残します (受信したプロパティーがすでに上書きされているため)。すべての -o 制限は、-x にも同様に適用されます。

zfs allow *filesystem* | *volume*
zfs allow [-ldug] everyone|*user*|*group*[,...] *perm* |@*setname*[,...] *filesystem*| *volume*
zfs allow [-ld] -e *perm*|@*setname*[,...] *filesystem* | *volume*
zfs allow -c *perm*|@*setname*[,...] *filesystem*|*volume*
zfs allow - s @*setname perm*|@*setname*[,...] *filesystem*|*volume*
zfs allow の構文および例についての完全な説明は、zfs_allow(1M) を参照してください。

zfs unallow [-rldug] everyone|*user*|*group*[,...] [*perm*|@*setname*[, ...]] *filesystem*| *volume*
zfs unallow [-rld] -e [*perm*|@*setname* [,...]] *filesystem*|*volume*
zfs unallow [- r] -c [*perm*|@*setname*[,...]]
filesystem|*volume*
zfs unallow [-r] -s @*setname* [*perm*|@*setname*[,...]]
filesystem|*volume*
zfs unallow の構文および例についての完全な説明は、zfs_allow(1M) を参照してください。

zfs hold [-r] *tag snapshot...*
tag 引数で指定された単一の参照を、指定された 1 つ以上のスナップショットに追加します。個々のスナップショットには独自のタグ名前空間があり、その空間内でタグが一意である必要があります。

スナップショットに保持が設定されている場合、保持されたそのスナップショットを zfs destroy コマンドを使用して破棄しようすると、EBUSY が返されます。

-r
指定されたタグを持つ保持をすべての子孫ファイルシステムのスナップショットに再帰的に適用することを指定します。

`zfs holds` [`-r`] *snapshot*...

指定された 1 つ以上のスナップショットについて、既存のユーザー参照をすべて一覧表示します。

`-r`

指定されたスナップショットでの保持を一覧表示するだけでなく、指定された子孫スナップショットで設定されている保持を一覧表示します。

`zfs release` [`-r`] *tag snapshot*...

tag 引数で指定された単一の参照を、指定された 1 つ以上のスナップショットから削除します。各スナップショットに対して、タグがすでに存在している必要があります。

スナップショットに保持が設定されている場合、保持されたそのスナップショットを `zfs destroy` コマンドを使用して破棄しようすると、`EBUSY` が返されます。

`-r`

指定されたタグを持つ保持をすべての子孫ファイルシステムのスナップショット上で再帰的に解除します。

`zfs diff` [`-FHte`] [`-o` *field*] ... *snapshot* [*snapshot* | *filesystem*]
`zfs diff -E` [`- FHt`] [`-o` *field*] ... *snapshot* | *filesystem*

スナップショットと子孫データセットとの差異に関する概要を表示します。子孫は、データセットのスナップショットか現在のデータセットのいずれかになります。

単一のスナップショットが指定された場合、そのスナップショットと現在のデータセットとの差異が表示されます。

元のスナップショットと子孫の間で変更があったファイルごとに、変更のタイプがファイルの名前とともに記述されます。名前の変更の場合、古い名前と新しい名前の両方が表示されます。ファイル名に含まれるスペース文字、バックスラッシュ文字、およびそのほかの非印字文字または非 7 ビット ASCII 文字は、バックスラッシュ文字のあとにそのバイト値を表す 3 桁の 8 進数が追加された形式で表示されます。

`-t` オプションが指定された場合、コマンドからの出力の最初の列はファイルの `st_ctim` 値になります。削除したファイルの場合、これは以前のスナップショットの最後の `st_ctim` になります。

変更のタイプはタイムスタンプのあとに表示され、単一の文字で記述されます。

`+` 以降のデータセットでファイルが追加されたことを示します。

`-` 以降のデータセットでファイルが削除されたことを示します。

`M` 以降のデータセットでファイルが変更されたことを示します。

`R` 以降のデータセットでファイルの名前が変更されたことを示します。

-F オプションが指定された場合、出力の次の列に、ファイルのタイプを示す単一の文字が表示されます。マッピングは次のとおりです。

F　通常ファイル

/　ディレクトリ

B　ブロック型デバイス

>　door

|　FIFO

@　シンボリックリンク

P　イベントポータル

=　socket

変更箇所に非ディレクトリファイルのリンク数の変更が含まれる場合、その変更は差分が括弧に囲まれた形式で変更行に表示されます。ファイル名が変更された場合、古いファイル名は文字列「->」を使用して新しいファイル名と分離されます。

-H オプションを選択すると、解析しやすい出力が生成されます。フィールドは単一のタブで区切られ、名前変更されたファイルの古い名前と新しい名前の間には矢印文字列 (->) は挿入されません。-H が選択されていない場合の出力では、フィールド間に空白が挿入されるとは限りません。

-e オプションが選択された場合、2 つのスナップショットの間で追加または変更されたすべてのファイルが列挙され、削除されたファイルは表示されません。変更タイプは修正の種類にかかわらず、常に「+」と報告されます。

-E オプションが選択された場合、空のスナップショットと指定されたスナップショットまたはデータセットとの差異として表示されます。

-o フィールドオプションが選択された場合、選択されたフィールドのみが表示されます。各行は、-F および -t オプションで要求された標準のフィールドで開始され、後続の -o オプションで要求されたフィールドが後に続きます。-H オプションの場合、すべてのフィールドが単一のタブで区切られます。次のフィールド名が許可されます。

object　ファイルについて ls -i によって出力される数値

parent　ファイルを含んでいるディレクトリについて ls -i によって出力される数値

size　ls -s によって表示されるファイルサイズ

links　ファイルへのリンク数

linkschange	ファイルへのリンク数の変更
name	ファイルの名前
oldname	名前を変更する前のファイルの名前。ファイルの名前が変更されていない場合は – (ハイフン)
user	ls によって表示されるファイルの所有者のユーザー名
group	ls によって表示されるファイルのグループ名
ctime	ファイルのメタデータが最後に変更されたときのタイムスタンプ
mtime	ファイルが最後に変更されたときのタイムスタンプ
atime	ファイルが最後にアクセスされたときのタイムスタンプ
crtime	ファイルが作成されたときのタイムスタンプ

{PRIV_SYS_CONFIG} または {PRIV_SYS_MOUNT} 特権がないユーザーがこのサブコマンドを使用するには、zfs allow で diff アクセス権を与えられる必要があります。

zfs key -l | {-a | [-r] *filesystem*|*volume*}
zfs key -u [-f] | {-a | [-r] *filesystem*|*volume*
zfs key -c [-o *keysource=value*] {-a | [-r] *filesystem*| *volume*}
zfs key -K {-a | [-r] *filesystem*|*volume*}

zfs key の構文および例についての完全な説明は、zfs_encrypt(1M) を参照してください。

使用例

例 1 ZFS ファイルシステム階層を作成する

次のコマンドは、pool/home という名前のファイルシステムと、pool/home/bob という名前のファイルシステムを作成します。マウントポイント /export/home が、親のファイルシステムに対して設定され、子のファイルシステムがそれを自動的に継承します。

```
# zfs create pool/home
# zfs set mountpoint=/export/home pool/home
# zfs create pool/home/bob
```

例 2 ZFS スナップショットを作成する

次のコマンドは、yesterday という名前のスナップショットを作成します。このスナップショットは、要求に応じて、pool/home/bob ファイルシステムのルートにある .zfs/snapshot ディレクトリにマウントされます。

```
# zfs snapshot pool/home/bob@yesterday
```

例 3　複数のスナップショットを作成および破棄する

次のコマンドは、pool/home とその子孫ファイルシステムの、yesterday という名前のスナップショットを作成します。各スナップショットは必要に応じて、そのファイルシステムのルートの .zfs/snapshot ディレクトリにマウントされます。2 つ目のコマンドは、新しく作成されたスナップショットを破棄します。

```
# zfs snapshot -r pool/home@yesterday
# zfs destroy -r pool/home@yesterday
```

例 4　ファイルシステムの圧縮を無効化および有効化する

次のコマンドは、pool/home 次のすべてのファイルシステムで compression プロパティーを無効にします。次のコマンドは、pool/home/anne に対して compression を明示的に有効にします。

```
# zfs set compression=off pool/home
# zfs set compression=on pool/home/anne
```

例 5　ZFS データセットを一覧表示する

次のコマンドは、システム内のアクティブなファイルシステムをすべて一覧表示します。スナップショットは、listsnaps プロパティーが on の場合に表示されます。デフォルトは off です。プールのプロパティーの詳細については、zpool(1M) を参照してください。

```
# zfs list
   NAME                     USED  AVAIL  REFER  MOUNTPOINT
   pool                     450K   457G    18K  /pool
   pool/home                315K   457G    21K  /export/home
   pool/home/anne            18K   457G    18K  /export/home/anne
   pool/home/bob            276K   457G   276K  /export/home/bob
```

例 6　ZFS ファイルシステムに割り当てを設定する

次のコマンドは、pool/home/bob に 30GB の割り当てを設定します。

```
# zfs set quota=30G pool/home/bob
```

例 7　ZFS プロパティーを一覧表示する

次のコマンドは、pool/home/bob のプロパティーをすべて一覧表示します。

```
# zfs get all pool/home/bob
NAME           PROPERTY              VALUE                  SOURCE
pool/home/bob  type                  filesystem             -
pool/home/bob  creation              Fri Jul 15 11:53 2011  -
pool/home/bob  used                  31K                    -
pool/home/bob  available             30.0G                  -
pool/home/bob  referenced            31K                    -
pool/home/bob  compressratio         1.00x                  -
pool/home/bob  mounted               yes                    -
```

例 7　ZFS プロパティーを一覧表示する　　(続き)

```
pool/home/bob  quota                 30G                    local
pool/home/bob  reservation           none                   default
pool/home/bob  recordsize            128K                   default
pool/home/bob  mountpoint            /pool/home/bob         default
pool/home/bob  sharenfs              off                    default
pool/home/bob  checksum              on                     default
pool/home/bob  compression           on                     default
pool/home/bob  atime                 on                     default
pool/home/bob  devices               on                     default
pool/home/bob  exec                  on                     default
pool/home/bob  setuid                on                     default
pool/home/bob  readonly              off                    default
pool/home/bob  zoned                 off                    default
pool/home/bob  snapdir               hidden                 default
pool/home/bob  aclmode               discard                default
pool/home/bob  aclinherit            restricted             default
pool/home/bob  canmount              on                     default
pool/home/bob  xattr                 on                     default
pool/home/bob  copies                1                      default
pool/home/bob  version               5                      -
pool/home/bob  utf8only              off                    -
pool/home/bob  normalization         none                   -
pool/home/bob  casesensitivity       mixed                  -
pool/home/bob  vscan                 off                    default
pool/home/bob  nbmand                off                    default
pool/home/bob  sharesmb              off                    default
pool/home/bob  refquota              none                   default
pool/home/bob  refreservation        none                   default
pool/home/bob  primarycache          all                    default
pool/home/bob  secondarycache        all                    default
pool/home/bob  usedbysnapshots       0                      -
pool/home/bob  usedbydataset         31K                    -
pool/home/bob  usedbychildren        0                      -
pool/home/bob  usedbyrefreservation  0                      -
pool/home/bob  logbias               latency                default
pool/home/bob  dedup                 off                    default
pool/home/bob  mlslabel              none                   default
pool/home/bob  sync                  standard               default
pool/home/bob  encryption            off                    -
pool/home/bob  keysource             none                   default
pool/home/bob  keystatus             none                   -
pool/home/bob  rekeydate             -                      default
pool/home/bob  rstchown              on                     default
pool/home/bob  shadow                none                   -
```

次のコマンドは、プロパティー値を 1 つ取得します。

例 7　ZFS プロパティーを一覧表示する　　(続き)

```
# zfs get -H -o value compression pool/home/bob
on
```

次のコマンドは、pool/home/bob のローカル設定を持つプロパティーをすべて一覧表示します。

```
# zfs get -r -s local -o name,property,value all pool/home/bob
NAME           PROPERTY              VALUE
pool/home/bob  quota                 30G
pool/home/bob  compression           on
```

例 8　ZFS ファイルシステムをロールバックする

次のコマンドは、pool/home/anne の内容を yesterday という名前のスナップショットに戻し、中間のスナップショットをすべて削除します。

```
# zfs rollback -r pool/home/anne@yesterday
```

例 9　ZFS クローンを作成する

次のコマンドは、初期内容が pool/home/bob@yesterday と同一の書き込み可能なファイルシステムを作成します。

```
# zfs clone pool/home/bob@yesterday pool/clone
```

例 10　ZFS クローンの移行を促す

次の各コマンドは、クローン、クローンの移行の促進、および名前変更を実行することで、あるファイルシステムに対する変更内容をテストしたあと、その変更後のファイルシステムで元のファイルシステムを置き換える方法を示しています。

```
# zfs create pool/project/production
  populate /pool/project/production with data
# zfs snapshot pool/project/production@today
# zfs clone pool/project/production@today pool/project/beta
make changes to /pool/project/beta and test them
# zfs promote pool/project/beta
# zfs rename pool/project/production pool/project/legacy
# zfs rename pool/project/beta pool/project/production
once the legacy version is no longer needed, it can be destroyed
# zfs destroy pool/project/legacy
```

例 11　ZFS プロパティーを継承する

次のコマンドにより、pool/home/bob および pool/home/anne は checksum プロパティーを親から継承します。

```
# zfs inherit checksum pool/home/bob pool/home/anne
```

例 12　ZFS データをリモートで複製する

次のコマンドは、リモートマシンにストリーム全体を送信してから、増分ストリームを送信して、poolB/received/fs@a および poolB/received/fs@b 内にそれぞれ復元します。poolB には、ファイルシステム poolB/received を含める必要があります。また、初期状態で poolB/received/fs を含んでいてはいけません。

```
# zfs send pool/fs@a | \
   ssh host zfs receive poolB/received/fs@a
# zfs send -i a pool/fs@b | ssh host \
   zfs receive poolB/received/fs
```

例 13　zfs receive -d オプションを使用する

次のコマンドは、poolA/fsA/fsB@snap のストリーム全体をリモートマシンに送信し、poolB/received/fsA/fsB@snap 内に受信します。受信するスナップショット名の fsA/fsB@snap 部分は、送信するスナップショットの名前に基づいて決定されます。poolB には、ファイルシステム poolB/received を含める必要があります。poolB/received/fsA が存在しない場合、空のファイルシステムとして作成されます。

```
# zfs send poolA/fsA/fsB@snap | \
   ssh host zfs receive -d poolB/received
```

例 14　ユーザープロパティーを設定する

次の例では、データセットのユーザー定義プロパティー com.example:department を設定します。

```
# zfs set com.example:department=12345 tank/accounting
```

例 15　ローリングスナップショットを実行する

次の例は、一貫性のある命名方式でスナップショットの履歴を維持する方法を示しています。1 週間分のスナップショットを保持するために、ユーザーは次のように、もっとも古いスナップショットを破棄し、残りのスナップショットの名前を変更してから、新しいスナップショットを作成します。

```
# zfs destroy -r pool/users@7daysago
# zfs rename -r pool/users@6daysago @7daysago
# zfs rename -r pool/users@5daysago @6daysago
# zfs rename -r pool/users@4daysago @5daysago
# zfs rename -r pool/users@3daysago @4daysago
# zfs rename -r pool/users@2daysago @3daysago
# zfs rename -r pool/users@yesterday @2daysago
# zfs rename -r pool/users@today @yesterday
# zfs snapshot -r pool/users@today
```

例 16　ZFS スナップショットの差異を表示する

次の例は、zfs diff -F および -t オプションを指定した場合の出力です。

例 16 ZFS スナップショットの差異を表示する (続き)

```
# zfs diff -Ft myfiles@snap1
1269962501.206726811   M   /   /myfiles/
1269962444.207369955   M   F   /myfiles/link_to_me      (+1)
1269962499.207519034   R       /myfiles/rename_me -> /myfiles/renamed
1269962431.813566720   -   F   /myfiles/delete_me
1269962518.666905544   +   F   /myfiles/new_file
1269962501.393099817   +   |   /myfiles/new_pipe
```

終了ステータス

次の終了値が返されます。

0
正常終了。

1
エラーが発生しました。

2
無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	確実

関連項目

chmod(1), chown(1), pktool(1), ssh(1), mount(1M), shadowd(1M), share(1M), share_nfs(1M), share_smb(1M), unshare(1M), zonecfg(1M), zpool(1M), chmod(2), chown(2), stat(2), write(2), fsync(3C), dfstab(4), vfstab(4), attributes(5)

gzip(1) のマニュアルページを参照してください。これは SunOS マニュアルページコレクションには含まれていません。

その他の ZFS 機能の詳細は、zfs_allow(1M)、zfs_encrypt(1M)、zfs_share(1M)、および『Oracle Solaris の管理: ZFS ファイルシステム』を参照してください。

注意事項

diff サブコマンドによって変更されたことが示されているファイルは、複数の方法で変更された可能性があります。変更箇所の報告は、st_ctim の変更の原因となったあらゆるアクション (stat(2) を参照) に基づいて行われます。

名前

zfs_allow – 特権のないユーザーに ZFS ファイルシステム管理アクセス権を委任する

形式

```
zfs allow filesystem|volume

zfs allow [-ldug] everyone|user|group[,...] perm|@setname[,...]
     filesystem|volume

zfs allow [-ld] -e perm|@setname[,...] filesystem|volume

zfs allow -c perm|@setname[,...] filesystem|volume

zfs allow -s @setname perm|@setname[,...] filesystem|volume

zfs unallow [-rldug] everyone|user|group[,...] [perm|@setname[,... ]]
     filesystem|volume

zfs unallow [-rld] -e [perm|@setname[,... ]] filesystem|volume

zfs unallow [-r] -c [perm|@setname[ ... ]] filesystem|volume

zfs unallow [-r] -s @setname [perm|@setname[,... ]] filesystem|volume
```

機能説明

zpool(1M) で説明されているように、`zfs allow` コマンドを使用して、ZFS ストレージプールで特権のないユーザーに ZFS ファイルシステムを管理するためのアクセス権を委任できます。`zfs unallow` コマンドを使用して、管理アクセス権を取り消すことができます。

一般にアクセス権とは、ZFS サブコマンドを使用する権限、または ZFS プロパティーを変更する権限です。使用できるアクセス権は次のとおりです。

NAME	TYPE	NOTES
allow	subcommand	Must also have the permission that is being allowed
clone	subcommand	Must also have the 'create' ability and 'mount' ability in the origin file system
create	subcommand	Must also have the 'mount' ability
destroy	subcommand	Must also have the 'mount' ability
diff	subcommand	Allows user to compare differences between snapshots of a ZFS dataset
hold	subcommand	Allows adding a user hold to a snapshot
key	subcommand	Allows key loading or unloading
keychange	other	Allows wrapping key change and re-key
mount	subcommand	Allows mount/umount of ZFS datasets
promote	subcommand	Must also have the 'mount' and 'promote' ability in the origin file system
receive	subcommand	Must also have the 'mount' and 'create' ability
release	subcommand	Allows releasing a user hold which might destroy the snapshot
rename	subcommand	Must also have the 'mount' and 'create' ability in the new parent
rollback	subcommand	
send	subcommand	
share	subcommand	Allows sharing file systems over NFS or

```
                                SMB protocols
snapshot         subcommand
groupquota       other          Allows accessing any groupquota@...
                                property
groupused        other          Allows reading any groupused@... property
userprop         other          Allows changing any user property
userquota        other          Allows accessing any userquota@... property
userused         other          Allows reading any userused@... property
aclinherit       property
aclmode          property
atime            property
canmount         property
casesensitivity  property
checksum         property
compression      property
copies           property
dedup            property
devices          property
encryption       property
exec             property
keysource        property
logbias          property
mlslabel         property
mountpoint       property
nbmand           property
normalization    property
primarycache     property
quota            property
readonly         property
recordsize       property
refquota         property
refreservation   property
reservation      property
rstchown         property
secondarycache   property
setuid           property
sharenfs         property
sharesmb         property
snapdir          property
sync             property
utf8only         property
version          property
volblocksize     property
volsize          property
vscan            property
xattr            property
zoned            property
```

サブコマンド

状態を変更するサブコマンドはすべて、元の形式でプールに永続的に記録されます。

`zfs ?`
ヘルプメッセージを表示します。

`zfs help` *command* | help | *property property-name* | *permission*
`zfs` コマンドの使用法情報を表示します。特定のコマンド、プロパティー、または委任アクセス権のヘルプを表示できます。特定のコマンドまたはプロパティーのヘルプを表示した場合、コマンド構文またはプロパティー値が表示されます。引数を何も付けずに `zfs help` を使用すると、`zfs` コマンドの完全なリストが表示されます。

`zfs allow` *filesystem* | *volume*
指定されたファイルシステムまたはボリューム上で委任されたアクセス権を表示します。詳細は、`zfs allow` のほかの形式を参照してください。

`zfs allow [-ldug] everyone|`*user*`|`*group*`[,...]` *perm*`|@`*setname*`[,...]` *filesystem*`|` *volume*
`zfs allow [-ld] -e` *perm*`|@`*setname*`[,...]` *filesystem* | *volume*
ファイルシステムに対する ZFS 管理アクセス権を非特権ユーザーに委任します。

`[-ug] everyone|`*user*`|`*group*`[,...]`
アクセス権を委任するユーザーを指定します。複数のエンティティーをコンマ区切りのリストとして指定できます。どちらの `-ug` オプションも指定されていない場合、引数はキーワード `everyone`、ユーザー名、最後にグループ名という優先順位で解釈されます。「everyone」という名前のユーザーまたはグループを指定するには、`-u` オプションまたは `-g` オプションを使用します。ユーザーと同じ名前のグループを指定するには、`-g` オプションを使用します。

`[-e]` *perm*`|@`*setname*`[,...]`
アクセス権を `everyone` に委任することを指定します。複数のアクセス権をコンマ区切りのリストとして指定できます。アクセス権の名前は、ZFS のサブコマンドおよびプロパティーの名前と同じです。下記のプロパティーリストを参照してください。プロパティーセットの名前を指定できます。プロパティーセット名はアットマーク記号 (@) で始まります。詳細については、下記の `-s` の形式を参照してください。

`[-ld]` *filesystem*`|`*volume*
アクセス権を委任する場所を指定します。`-ld` オプションのどちらも指定されていない場合や、両方が指定されている場合は、ファイルシステムまたはボリュームおよびそのすべての子孫にアクセス権が許可されます。`-l` オプションのみが指定されている場合は、指定されたファイルシステムのみに「ローカルの」アクセス権が許可されます。`-d` オプションのみが指定されている場合は、子孫ファイルシステムのみにアクセス権が許可されます。

`zfs allow -c` *perm*`|@`*setname*`[,...]` *filesystem*`|`*volume*
「作成時」アクセス権を設定します。これらのアクセス権は、新しく作成された子孫ファイルシステムの作成者に (ローカルに) 付与されます。

zfs allow -s @*setname perm*|@*setname*[,...] *filesystem*|*volume*
アクセス権セットにアクセス権を定義または追加します。このセットは、指定のファイルシステムとその子孫に対してほかの zfs allow コマンドで使用できます。セットは動的に評価されるため、セットに加えられた変更はすぐに反映されます。アクセス権セットは ZFS ファイルシステムと同じ命名規則に従いますが、名前はアットマーク記号 (@) で始まり、64 文字以下の長さにする必要があります。

zfs unallow [-rldug] everyone|*user*|*group*[,...] [*perm*|@*setname*[, ...]] *filesystem*| *volume*
zfs unallow [-rld] -e [*perm*|@*setname* [,...]] *filesystem*|*volume*
zfs unallow [- r] -c [*perm*|@*setname*[,...]]
filesystem|*volume*
zfs allow コマンドで付与されたアクセス権を削除します。どのアクセス権も明示的には拒否されないため、付与されているほかのアクセス権は引き続き有効です。たとえば、祖先によってアクセス権が付与されているとします。アクセス権が指定されていない場合は、指定された *user*、*group*、または everyone のすべてのアクセス権が削除されます。everyone を指定すると (または、-e オプションを使用すると)、everyone に付与されていたアクセス権のみが削除されます。すべてのユーザーおよびグループのすべてのアクセス権が削除されるわけではありません。zfs allow コマンドの -ldugec オプションの説明を参照してください。

-r
このファイルシステムおよびすべての子孫から、アクセス権を再帰的に削除します。

zfs unallow [-r] -s @*setname* [*perm*|@*setname*[,...]]
filesystem|*volume*
アクセス権セットからアクセス権を削除します。アクセス権が指定されていない場合は、すべてのアクセス権が削除されるため、セット全体が削除されます。

使用例

例 1　ZFS データセットの ZFS 管理アクセス権を委任する

次の例は、ユーザー anne が pool/home/anne の作成、破棄、マウント、およびスナップショットの取得を行えるようにアクセス権を設定する方法を示しています。pool/home/anne に対するアクセス権も表示されています。

```
# zfs allow anne create,destroy,mount,snapshot pool/home/anne
# zfs allow pool/home/anne
---- Permissions on pool/home/anne -----------------------------------
Local+Descendent permissions:
        user anne create,destroy,mount,snapshot
```

pool/home/anne マウントポイントのアクセス権はデフォルトで 755 に設定されているため、ユーザー anne はファイルシステムを pool/home/anne にマウントできません。次の構文と同様の ACL を設定して、マウントポイントにアクセスできるようにします。

```
# chmod A+user:anne:add_subdirectory:allow /pool/home/anne
```

例 2 ZFS データセットの作成時アクセス権を委任する

次の例は、pool/home にファイルシステムを作成するためのアクセス権をグループ staff のメンバーに付与する方法を示しています。この構文では、staff のメンバーが自分のファイルシステムを破棄することは許可し、ほかのユーザーのファイルシステムを破棄することは許可していません。pool/home に対するアクセス権も表示されています。

```
# zfs allow staff create,mount pool/home
# zfs allow -c destroy pool/home
# zfs allow pool/home
---- Permissions on pool/home ----------------------------------------
Create time permissions:
        destroy
Local+Descendent permissions:
        group staff create,mount
```

例 3 ZFS データセットのアクセス権セットを定義して付与する

次の例は、pool/home ファイルシステムのアクセス権セットを定義して付与する方法を示しています。pool/home に対するアクセス権も表示されています。

```
# zfs allow -s @pset create,destroy,snapshot,mount pool/home
# zfs allow staff @pset pool/home
# zfs allow pool/home
---- Permissions on pool/home ----------------------------------------
Permission sets:
        @pset create,destroy,mount,snapshot
Create time permissions:
        destroy
Local+Descendent permissions:
        group staff @pset,create,mount
```

例 4 ZFS データセットのプロパティーアクセス権を委任する

次の例は、tank/users ファイルシステムに割り当てと予約を設定する能力を付与する方法を示しています。tank/users に対するアクセス権も表示されています。

```
# zfs allow mark quota,reservation tank/users
# zfs allow tank/users
---- Permissions on tank/users ---------------------------------------
Local+Descendent permissions:
        user mark quota,reservation
mark% zfs set quota=10G tank/users/tim
mark% zfs get quota tank/users/tim
NAME            PROPERTY  VALUE  SOURCE
tank/users/tim  quota     10G    local
```

例 5　ZFS データセットの ZFS 委任アクセス権を削除する

次の例は、pool/home ファイルシステムの staff グループについて、@pset アクセス権セットからスナップショットアクセス権を削除する方法を示しています。pool/home に対するアクセス権も表示されています。

```
# zfs unallow -s @pset snapshot pool/home
# zfs allow pool/home
---- Permissions on pool/home --------------------------------------
Permission sets:
        @pset create,destroy,mount
Create time permissions:
        destroy
Local+Descendent permissions:
        group staff @pset,create,mount
```

終了ステータス

次の終了値が返されます。

0
　正常終了。

1
　エラーが発生した。

2
　無効なコマンド行オプションが指定された。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	確実

関連項目

zfs(1M), zpool(1M), chmod(2), chown(2), attributes(5)

その他の ZFS 機能の詳細は、zfs_encrypt(1M)、zfs_share(1M)、zfs(1M)、および『Oracle Solaris の管理: ZFS ファイルシステム』を参照してください。

名前

zfs_encrypt – ZFS ファイルシステムを暗号化する

形式

```
zfs [-?]

zfs help subcommand | help | property property-name | permission

zfs create -o encryption=on  [-o keysource=raw | hex | passphrase,prompt | file://|pkcs11:|https://] ..

zfs clone [-p] [-K] [-o property=value] ... snapshot filesystem|volume

zfs get [-r|-d depth][-Hp][-o all | field[,...]] [-s source[,...]]
     all | property[,...] filesystem|volume|snapshot ...

zfs key-l | {-a | [-r] filesystem|volume}

zfs key -u [-f] {-a | [-r] filesystem|volume}

zfs key -c [-o keysource=value] {-a | [-r] filesystem|volume}

zfs key -K {-a | [-r] filesystem|volume}

zfs mount

zfs mount [-vO] [-o options] -a | filesystem

zfs unmount [-f] -a | filesystem|mountpoint
```

機能説明

zpool(1M) で説明されているように、`zfs create -o encryption` コマンドは、ZFS ストレージプール内で新しく作成された ZFS データセットを暗号化します。

暗号化

暗号化とは機密保護のためにデータをエンコードするプロセスで、エンコードされたデータにデータ所有者がアクセスするには鍵が必要になります。暗号化ポリシーは ZFS データセットの作成時に設定できますが、ポリシーの変更はできません。詳細については、「ネイティブプロパティー」の節の `encryption` および `keysource` プロパティーの説明を参照してください。

データセットの暗号化は恒久的に継承され、データセットのクローン作成中に削除することはできません。複製されたデータセットストリームを受信したときに、暗号化が必要な場合は、宛先データセットで暗号化が有効になっている必要があります。そうでない場合、データは平文で格納されます。暗号化データセットの完全に複製されたストリームにより、新しく生成された鍵の下での暗号化データセットが生じますが、ストリーム自体は暗号化されません。

ネイティブな ZFS 暗号化プロパティー

ZFS 暗号化に関連する次のネイティブなプロパティーは、データセットに関する読み取り専用の統計情報で構成されています。これらのプロパティーは、設定も継承もできません。ネイティブプロパティーは、特に注記がないかぎり、すべてのデータセットタイプに適用されます。ZFS ネイティブプロパティーの完全な説明およびリストは、zfs(1M) を参照してください。

`keystatus`
: データセットの暗号化鍵のステータスを識別します。データセットの鍵を利用できるかどうかは、`available` または `unavailable` のステータスで示されます。暗号化が有効になっていないデータセットの場合、`none` が表示されます。

mounted
: ファイルシステムの場合は、ファイルシステムが現在マウントされているかどうかを示します。このプロパティーは、yes または no になります。

rekeydate
: このデータセットに対する zfs key -K または zfs clone -K 操作によりデータ暗号化鍵を最後に変更した日付。鍵の変更操作が実行されていない場合、rekeydate は creation と同じ日付になります。

次のプロパティーは、ファイルシステムの作成後には変更できないため、ファイルシステムの作成時に設定するようにしてください。zfs create または zpool create コマンドで設定されていない場合、これらのプロパティーは親データセットから継承されます。これらの機能がサポートされる前に親データセットが作成されたため、親データセットにこれらのプロパティーが存在しない場合、新しいファイルシステムではこれらのプロパティーにデフォルト値が設定されます。

encryption=off | on | aes-128-ccm | aes-192-ccm | aes-256-ccm | aes-128-gcm | aes-192-gcm | aes-256-gcm
: 暗号化されたデータセットに使用する暗号化アルゴリズムと鍵長を定義します。on の値は aes-128-ccm と同じです。デフォルト値は off です。encryption が off 以外の値に設定されている場合、checksum プロパティーは sha256+mac に設定され、readonly になります。

次のプロパティーは、作成時に指定する必要があり、特殊コマンドを使用すると変更できます。

keysource=*raw* | *hex* | *passphrase*,*prompt* | *file*:// | pkcs11: | https://
: データセットの鍵をラップする鍵の形式と場所を定義します。zfs key -l コマンドを使用してデータセットを作成、マウント、または読み込むときに、この鍵が必要になります。

  keysource プロパティーには、鍵の表示方法を指定する format と、鍵の取得先を特定する locator の 2 つの値を指定できます。

  format には次の 3 つの値を指定できます。

  - *raw*: 生の鍵バイト
  - *hex*: 16 進数の鍵文字列
  - *passphrase*: 鍵を生成する文字列

  locator には次の 2 つの値を指定できます。

  - prompt: データセットの作成またはマウント時に鍵の入力を要求
  - file:///*filename*: ファイルシステム内の鍵ファイルの場所
  - pkcs11: PKCS#11 トークンでの鍵の場所を記述した URI
  - https://*location*: 安全なサーバー上の鍵ファイルの場所

  https:// のロケータを使用して鍵を作成する例については、「使用例」を参照してください。

ラップ鍵の値または鍵を変更するには、zfs key -c コマンドを実行する必要があります。鍵の場所のみを変更する (たとえば、ファイル名を変更する) 必要がある場合は、zfs set コマンドを keysource プロパティーとともに使用します。zfs set コマンドで鍵の場所だけが変更された場合、新しい場所は有効なラップ鍵を持っているかどうかなどの ZFS によるチェックは行われません。

keysource が指定も継承もされていない場合、デフォルトの keysource は、暗号化が有効なデータセットでは passphrase,prompt に、暗号化が無効なデータセットでは none に設定されます。

サブコマンド

状態を変更するサブコマンドはすべて、元の形式でプールに永続的に記録されます。

zfs ?
ヘルプメッセージを表示します。

zfs help *command* | help | *property property-name* | *permission*
zfs コマンドの使用法情報を表示します。特定のコマンド、プロパティー、または委任アクセス権のヘルプを表示できます。特定のコマンドまたはプロパティーのヘルプを表示した場合、コマンド構文またはプロパティー値が表示されます。引数を何も付けずに zfs help を使用すると、zfs コマンドの完全なリストが表示されます。

zfs create [-p] [-o *encryption=on*] [-o keysource=*raw* | *hex* | *passphrase*, *prompt* | *file*://|pkcs11:|https://] ... *filesystem*
aes-128-ccm を使用する、暗号化が有効になった新しい ZFS ファイルシステムを作成します。サポートされている暗号化アルゴリズムのリストについては、暗号化プロパティーの説明を参照してください。

-p
存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した mountpoint プロパティーに従って自動的にマウントされます。コマンド行で -o オプションを使用して指定したプロパティーはすべて無視されます。ターゲットのファイルシステムがすでに存在する場合、処理は正常に完了します。

-o *encryption=value*
暗号化プロパティーを *value* に設定します。複数の -o オプションを指定できます。複数の -o オプション内で同じプロパティーを指定した場合は、エラーが発生します。

zfs clone [-p] [-K] [-o *property=value*] ... *snapshot filesystem*|*volume*
指定したスナップショットのクローンを作成します。詳細は、「クローン」の節を参照してください。ターゲットのデータセットは、ZFS 階層内の任意の場所に配置できます。作成されたデータセットは元のデータセットと同タイプになります。

-p
存在しない親データセットをすべて作成します。この方法で作成されたデータセットは、それぞれの親から継承した mountpoint プロパティーに従って自動的にマウントされます。ターゲットのファイルシステムまたはボリュームがすでに存在する場合、処理は正常に完了します。

-o *property=value*
指定されたプロパティーを設定します。詳細は、zfs create を参照してください。

-K
このデータセットの鍵チェーンに新しいデータ暗号化鍵を作成します。クローンに書き込まれたデータは、新しいデータ暗号化鍵を使用します。これは元のスナップショットとは異なる鍵です。

zfs set keysource=*value filesystem*|*volume*| ...
keysource プロパティーを、各データセット用に指定した値に設定します。keysource の場所のみを変更できます。ラップ鍵の値を変更する場合は、zfs key -c コマンドを使用します。

-r
子データセットのサブツリー全体に対して、設定の有効な値を再帰的に適用します。有効な値は、プロパティーに応じて設定または継承できます。

zfs get encryption | keysource | keystatus | rekeydate *filesystem*|*volume*| ...
指定したデータセットのプロパティーを表示します。

-r
任意の子のプロパティーを再帰的に表示します。

-d *depth*
データセットのすべての子を再帰的に表示しますが、再帰の深さは *depth* に制限されます。深さ 1 の場合は、データセットとその直接の子だけが表示されます。

-H
スクリプトによる解析がより容易な形式で、出力を表示します。ヘッダーがすべて省略され、フィールドが任意の数のスペースではなく、タブ 1 つで明示的に区切られます。

zfs key-l | {-a | [-r] *filesystem*|*volume*}
1 つのデータセットと、その鍵を継承するすべてのデータセットのための暗号化鍵を読み込みます。このコマンドで指定される鍵は、データセットの暗号化に使用される実際の鍵ではありません。データセットのデータ暗号化鍵のセットのためのラップ鍵です。

-l
暗号化されたデータセットとその鍵を継承するデータセットをロック解除するラップ鍵を読み込みます。このコマンドは、データセットの keysource プロパティーで定義された内容に基づいて鍵を読み込みます。

プールのインポート中、鍵の読み込み操作は、データセットのマウント時に実行されます。起動中、ラップ鍵が使用可能で、keysource が prompt に設定されていない場合に、鍵読み込み操作は実行されます。

-a
システム上のすべてのプール内のすべてのデータセットに適用します。

-r
指定されたファイルシステムまたはボリュームの下にあるすべてのデータセットに再帰的に操作を適用します。

zfs key-u [-f] | {-a | [-r] *filesystem*|*volume*}
1 つのデータセットとその鍵を継承するすべてのデータセットのための暗号化鍵を読み込み解除します。

-u
データセットをマウント解除してから、1 つの暗号化されたデータセットとその鍵を継承するデータセットのためのラップ鍵の読み込み解除を試みます。成功した場合、データセットは、アクセスができなくなり、マウントが解除されます。

-f
鍵の読み込み解除を試みる前に、データセットを強制的にマウント解除しようとします。指定しない場合、通常のマウント解除が試みられます。

-a
システム上のすべてのプール内のすべてのデータセットに適用します。

-r
指定されたファイルシステムまたはボリュームの下にあるすべてのデータセットに再帰的に操作を適用します。

zfs key-c [-o keysource=*value*] | {-a | [-r] *filesystem*|*volume*}
ラップ鍵を変更します。新しい鍵の形式またはロケータが異なる場合、keysource プロパティーをコマンドの一部に含める必要があります。zfs key -c コマンドの一部として変更できるのは keysource プロパティーだけです。

-c
1 つの暗号化されたデータセットの鍵とそれを継承するデータセットのためのラップ鍵を変更します。鍵の変更操作を行う前に、既存の鍵がすでに読み込まれている必要があります。ZFS は、既存のパスフレーズの入力を要求しません。

-o *property=value*
鍵変更操作の一部として変更するプロパティー。keysource プロパティーは、鍵変更操作の一部として変更できる唯一のオプションです。

keysource プロパティーを変更する権限が必要になります。

-a
システム上のすべてのプール内のすべてのデータセットに適用します。

-r
指定されたファイルシステムまたはボリュームの下にあるすべてのデータセットに再帰的に操作を適用します。

zfs key-K {-a | [-r] *filesystem*|*volume*}
新しいデータ暗号化鍵を作成します。新しいデータ暗号化鍵は、このデータセットのすべての既存データ暗号化鍵と同じラップ鍵でラップされます。

-K
このデータセットの新しいデータ暗号化鍵を作成します。この操作後に書き込まれたデータは、新しいデータ暗号化鍵を使用します。

-a
システム上のすべてのプール内のすべてのデータセットに適用します。

-r
指定されたファイルシステムまたはボリュームの下にあるすべてのデータセットに再帰的に操作を適用します。

zfs mount
zfs mount [-vO] [-o *options*] -a | *filesystem*
ZFS ファイルシステムをマウントします。これは、ブートプロセスの一部として自動的に呼び出されます。zfs mount の構文についての完全な説明は、zfs(1M) を参照してください。

filesystem
指定されたファイルシステムをマウントします。

暗号化されたデータセットの zfs mount 操作では、keysource プロパティー値によっては鍵の入力が求められる場合があります。これは、たとえば keysource ロケータが prompt に設定されている場合に起こることがあります。

zfs unmount [-f] -a | *filesystem*|*mountpoint*
現在マウントされている ZFS ファイルシステムをマウント解除します。これは、シャットダウンプロセスの一部として自動的に呼び出されます。zfs unmount の構文についての完全な説明は、zfs(1M) を参照してください。

filesystem|*mountpoint*
指定されたファイルシステムをマウント解除します。このコマンドには、システム上の ZFS ファイルシステムのマウントポイントのパスを指定することもできます。

暗号化されたデータセットの場合、ファイルシステムがマウント解除されているときには、鍵の読み込みは解除されません。鍵を読み込み解除するには、zfs key を参照してください。

使用例

例 1 暗号化されたデータセットを作成する

次の例は、keysource プロパティーのデフォルト値である passphrase プロンプトを使用して、暗号化されたデータセットを作成する方法を示しています。この例では、tank/home データセットは暗号化されていないものとします。

```
# zfs create -o encryption=on tank/home/bob
Enter passphrase for 'tank/home/bob/': **********
Enter again: **********
```

次の例では、pktool(1) コマンドを使用して生の鍵をファイルに生成しています。次に、aes-256-ccm アルゴリズムと、pktool で生成された生の鍵のファイルを使用して、暗号化されたデータセット (tank/home/anne) を作成しています。

```
# pktool genkey keystore=file outkey=/media/stick/mykey \
keytype=aes keylen=256
# zfs create encryption=aes-256-ccm \
-o keysource=raw,file:///rmdisk/stick/mykey tank/home/anne
```

この例は、http の場所に格納されている鍵を要求する、暗号化された ZFS ファイルシステムを作成する方法を示しています。

```
# zfs create -o encryption=on \
-o keysource=passphrase,https://keys.example.com/keys/42 tank/home/fs1
```

この例は、PKCS#11 トークン内で生の鍵を生成する方法を示しています。次に、pktool から生成された生の PKCS#11 鍵を使用して、暗号化されたデータセットを作成します。

```
# pktool genkey keystore=pkcs11 keytype=aes keylen=128 label=fs2
Enter PIN for Sun Software PKCS#11 softtoken: xxxxx
# zfs create -o encryption=on -o keysource=raw,pkcs11:object=fs2 \
tank/home/fs2
Enter PKCS#11 token PIN for 'tank/home/fs2': xxxxx
```

この例は、KMS トークン内で生の鍵を生成する方法を示しています。次に、pktool から生成された生の KMS 鍵を使用して、暗号化されたデータセットを作成します。

```
# pktool genkey keystore=pkcs11 keytype=aes keylen=256 token=KMS \
label=fs3
Enter PIN for KMS: xxxxx
# zfs create -o encryption=aes-256-ccm \
-o keysource="raw,pkcs11:token=KMS;object=fs3" tank/home/fs3
Enter 'KMS' PKCS#11 token PIN for 'tank/home/fs3': xxxxx
```

例 2 別の暗号化アルゴリズムを使用して、暗号化されたデータセットを作成する

この例では、すべての tank/home データセットが keysource プロパティーを継承しますが、tank/home/bob データセットは別の暗号化アルゴリズムを使用して作成されます。

例 2 別の暗号化アルゴリズムを使用して、暗号化されたデータセットを作成する (続き)

```
# zpool create tank ....
# zfs create -o encryption=on tank/home
# zfs get keystatus tank/home
NAME       PROPERTY   VALUE        SOURCE
tank/home  keystatus  available    -

# zfs create -o encryption=aes-256-ccm tank/home/bob
```

例 3 encryption プロパティーおよび keysource プロパティーを継承する

この例では、すべての tank/home データセットが、encryption プロパティーおよび keysource プロパティーを継承します。

```
# zpool create -o encryption=on -o keysource=raw,file:///... tank ...
# zfs create tank/home
```

例 4 暗号化されたデータセットのラップ鍵および鍵ソースを変更する

次の例は、データセットのラップ鍵を、そのデータセットの keysource プロパティーで定義された新しい鍵に変更する方法を示しています。

```
# zfs get keysource tank/home/bob
NAME  PROPERTY         VALUE                    SOURCE
tank  keysource        raw,file:///etc/keyfile  default

# zfs key -c -o keysource=passphrase,prompt  tank/home/bob
Enter passphrase for 'tank/home/bob/': **********
Enter again: **********
```

次の例は、データセットのラップ鍵の http の場所を変更する方法を示しています。

```
# zfs get keysource tank/home/bob
NAME           PROPERTY   VALUE              SOURCE
tank/home/bob  keysource  passphrase,prompt  local

# zfs set keysource=passphrase,https://internal.example.com/keys/bob/zfs \
tank/home/bob
```

keysource プロパティーを変更するには、委任された key アクセス権および keychange アクセス権が必要です。

例 5 データセットの暗号化鍵を変更する

次の例は、ユーザーおよび管理者が表示も管理もできないデータセットの暗号化鍵を変更する方法を示しています。データセットの暗号化鍵は、keysource プロパティーで指定された鍵でラップ (暗号化) されています。

例 5 データセットの暗号化鍵を変更する (続き)

```
# zfs key -K tank/project42
# zfs get rekeydate,creation tank/project42
```

鍵変更操作を実行するには、委任された keychange アクセス権が必要です。

終了ステータス

次の終了値が返されます。

0
正常終了。

1
エラーが発生した。

2
無効なコマンド行オプションが指定された。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	確実

関連項目

chmod(1), chown(1), pktool(1), ssh(1), mount(1M), zfs(1M), zpool(1M), chmod(2), chown(2), stat(2), write(2), attributes(5)

その他の ZFS 機能の詳細は、zfs_allow(1M)、zfs_share(1M)、zfs(1M)、および『Oracle Solaris の管理: ZFS ファイルシステム』を参照してください。

名前

zfs_share – ZFS ファイルシステムを共有または共有解除する

形式

```
zfs [-?]

zfs help subcommand | help | property property-name | permission

zfs create [-p] [-o sharenfs=on | sharesmb=on] [-o property ... filesystem

zfs set share=name=sharename,path=pathname,[desc=description],
prot=smb | nfs[,property=value[,...]] filesystem

zfs  set [-r] sharenfs=on | off filesystem

zfs  set [-r] sharesmb=on | off filesystem

zfs set -c share=name=sharename filesystem

zfs get [-r] sharenfs filesystem ...

zfs  get [-r] sharesmb filesystem ...

zfs  get  [-r] share filesystem ...

zfs share -a | filesystem

zfs unshare -a filesystem|mountpoint
```

機能説明

zpool(1M) で説明されているように、zfs set share コマンドを使用して ZFS ストレージプール内に NFS または SMB 共有を作成します。sharenfs または sharesmb をオンに設定すると、共有を公開します。また、zfs share および zfs unshare コマンドを使用すると、ZFS データセットの NFS 共有を公開および非公開にできます。

ネイティブな共有ファイルシステムプロパティー

ファイルシステムプロパティーは、ネイティブプロパティーとユーザー定義 (「ユーザー」) プロパティーの 2 種類に分けられます。ネイティブプロパティーは、情報を表示するか ZFS の動作を制御します。また、ネイティブプロパティーは編集可能であるか、読み取り専用です。

プロパティーは、子により上書きされないかぎり、親から継承されます。一部のプロパティーは、特定のタイプのデータセット (ファイルシステム、ボリューム、またはスナップショット) にのみ適用されます。

次のネイティブプロパティーを使用すると ZFS ファイルシステムの動作を変更することが可能であり、通常ファイルシステムの共有時に使用されます。

次の NFS および SMB ネイティブプロパティーを使用すると、ZFS 共有の動作を変更できます。

nbmand=on|off
: ファイルシステムを nbmand (非ブロッキング強制ロック) でマウントするべきかどうかを制御します。これは SMB クライアントに使用されます。このプロパティーの変更は、ファイルシステムをマウント解除してから再マウントしたときにはじめて反映されます。nbmand マウントの詳細は、mount(1M) を参照してください。

readonly=on | off
このデータを変更できるかどうかを制御します。デフォルト値は off です。

このプロパティーは、列の短縮名 rdonly を使用して参照することもできます。

sharenfs=on | off
ZFS データセットを NFS 共有として公開するかどうかを制御します。また、zfs share および zfs unshare コマンドを使用すると、ZFS データセットの NFS 共有を公開および非公開にできます。どちらの NFS 共有公開方法でも、NFS 共有プロパティーがすでに設定されている必要があります。NFS 共有プロパティーの設定の詳細については、zfs set share コマンドを参照してください。

sharenfs プロパティーを変更すると、ファイルシステム共有、およびそのプロパティーを継承するすべての子は、そのプロパティーが以前に off であった場合、またはそのプロパティーの変更前に共有が公開されていた場合にのみ、zfs set share コマンドで設定されている新しいオプションで再度公開されます。新しいプロパティー値が off であれば、ファイルシステム共有は非公開になります。

sharesmb=on | off
ZFS データセットを SMB 共有として公開するかどうかを制御します。また、zfs share および zfs unshare コマンドを使用すると、ZFS データセットの SMB 共有を公開および非公開にできます。どちらの SMB 共有公開方法でも、SMB 共有プロパティーが同時に設定されている必要があります。SMB 共有プロパティーの設定の詳細については、zfs set share コマンドを参照してください。

SMB 共有にはリソース名が必要であるため、データセット名から一意のリソース名が構築されます。構築される名前はデータセット名のコピーですが、リソース名では使用できない文字がデータセット名に含まれている場合、それらは下線 (_) で置き換えられます。データセット名を指定の名前で置き換えるための擬似プロパティー「name」もサポートされています。この指定された名前は、継承の際に先頭部分のデータセットを置き換えるために使用されます。たとえば、データセット data/home/john が name=john と設定されている場合、data/home/john のリソース名は john になります。john に data/home/john/backups という子データセットがある場合、そのリソース名は john_backups になります。

SMB 共有が作成されると、SMB 共有名が .zfs/shares ディレクトリのエントリとして表示されます。ls または chmod コマンドを使用して、このディレクトリのエントリについて共有レベルの ACL を表示できます。

sharesmb プロパティーを変更すると、データセット共有、およびそのプロパティーを継承するすべての子は、そのプロパティーが以前に off に設定されていた場合、またはそのプロパティーの変更前に共有が公開されていた場合に、zfs set share コマンドで設定されている新しいオプションで再度公開されます。新しいプロパティー値が off であれば、データセット共有は非公開になります。

vscan=on|off
: 通常ファイルを開くときや閉じるときに、ファイルのウイルススキャンを実行するかどうかを制御します。ウイルススキャンを実行するには、このプロパティーを有効にするほかに、ウイルススキャンサービスも有効にする必要があります。デフォルト値は off です。

次のプロパティーは、ファイルシステムの作成後には変更できないため、ファイルシステムの作成時に設定するようにしてください。zfs create または zpool create コマンドで設定されていない場合、これらのプロパティーは親データセットから継承されます。これらの機能がサポートされる前に親データセットが作成されたため、親データセットにこれらのプロパティーが存在しない場合、新しいファイルシステムではこれらのプロパティーにデフォルト値が設定されます。

casesensitivity = sensitive | insensitive | mixed
: ファイルシステムで使用するファイル名照合アルゴリズムで、大文字と小文字を区別するか、区別しないか、または両方の照合方式の組み合わせを許可するかを指定します。casesensitivity プロパティーのデフォルト値は mixed です。通常、UNIX および POSIX のファイルシステムでは、ファイル名の大文字と小文字を区別します。

  casesensitivity プロパティーの値が mixed の場合、ファイルシステムが要求に応じて大文字と小文字を区別する照合も区別しない照合もサポートできることを示します。現在、混合動作をサポートするファイルシステムで大文字と小文字を区別しない照合が可能なのは、Solaris SMB サーバー製品に限られています。mixed 値の動作の詳細については、『Oracle Solaris の管理: ZFS ファイルシステム』を参照してください。

normalization = none | formC | formD | formKC | formKD
: ファイルシステムで 2 つのファイル名を比較するとき常にファイル名の unicode 正規化を実行するかどうか、およびどの正規化アルゴリズムを使用するかを指定します。保存されるファイル名が変更されることはなく、名前の正規化は比較処理の一部として実行されます。このプロパティーが none 以外の有効な値に設定されており、utf8only プロパティーが指定されなかった場合、utf8only プロパティーは自動的に on に設定されます。normalization プロパティーのデフォルト値は none です。このプロパティーは、ファイルシステムの作成後には変更できません。

utf8only =on | off
: UTF-8 文字コードセットに存在しない文字が含まれているファイル名をファイルシステムで拒否するかどうかを設定します。このプロパティーが明示的に off に設定されている場合、normalization プロパティーを明示的に設定しないか、または none に設定する必要があります。utf8only プロパティーのデフォルト値は off です。このプロパティーは、ファイルシステムの作成後には変更できません。

サブコマンド

状態を変更するサブコマンドはすべて、元の形式でプールに永続的に記録されます。

`zfs ?`
ヘルプメッセージを表示します。

`zfs help` *command* | help | *property property-name* | *permission*
`zfs` コマンドの使用法情報を表示します。特定のコマンド、プロパティー、または委任アクセス権のヘルプを表示できます。特定のコマンドまたはプロパティーのヘルプを表示した場合、コマンド構文またはプロパティー値が表示されます。引数を何も付けずに `zfs help` を使用すると、`zfs` コマンドの完全なリストが表示されます。

`zfs create` [`-p`] [`-o sharenfs=on | sharesmb=on -o`*property*=*value*] ... *filesystem*
新しい ZFS ファイルシステムを作成します。ファイルシステムは、親から継承した `mountpoint` プロパティーに従って自動的にマウントされます。

`-o` *property*=*value*
データセットの作成時にコマンド `zfs set` *property*=*value* が呼び出された場合と同様に、指定されたプロパティーを設定します。編集可能なすべての ZFS プロパティーは、作成時にも設定可能です。複数の `-o` オプションを指定できます。複数の `-o` オプション内で同じプロパティーを指定した場合は、エラーが発生します。

`zfs get [-r] share` [*filesystem*]
定義されているすべての共有、または指定されたファイルシステムか子孫のファイルシステム用に定義されている共有を表示します。

`zfs set share=`*name*=*sharename*,`path=`*pathname*,[`desc=`*description*],`prot=smb` | `nfs`[,*property*=*value*[,...]] *filesystem*
ZFS データセットの NFS または SMB ファイル共有プロパティーを定義します。NFS または SMB ファイル共有プロパティーの設定後、`sharenfs` または `sharesmb` プロパティーを `on` に設定することによって、ZFS ファイルシステムの NFS または SMB 共有を公開できます。

次の共有プロパティーは必須です。

`share=`*name*=*sharename*
NFS または SMB 共有名を指定します。共有名の最大長は、80 文字です。

`path=`*pathname*
共有されるデータセットの物理パス、または共有されるデータセット内のサブディレクトリの物理パスを指定します。

`prot=smb` | `nfs`
NFS または SMB ファイル共有プロトコルを識別します。

次の共有プロパティーは省略可能です。

desc=*description*
共有リソースを表すテキスト文字列を指定します。説明内のスペースまたはコンマは、引用符 (“ “) で囲む必要があります。

ro=|rw=
すべてのクライアントに対して共有を読み取り/書き込み可能にするか読み取り専用にするかを指定します。ホスト名、IP アドレス、またはネットグループを含むコロン区切りのリストを指定することもできます。

root=
ルートアクセスを許可された特定のホストまたはホストのリストから root ユーザーを指定します。デフォルトでは、ルートアクセスを許可されたホストはありません。

sec=
sys、dh、krb5 などの NFS サーバーセキュリティーモードを指定します。サポートされているセキュリティーモード情報については、nfssec(5) を参照してください。

次の省略可能な NFS プロパティーは、prot=nfs のあとで、sec プロパティーの前に指定する必要があります。

```
anon=TYPE_USER
nosub=[true|false]
nosuid=[true|false]
aclok=[true|false]
public=[true|false]
index=filename
log=TYPE_LOGTAG
cksum=TYPE_STRINGSET
```

次の省略可能な SMB プロパティーは、prot=smb のあとに指定する必要があります。

```
ad-container=string
abe=[true|false]
csc=[disabled|manual|auto|vdo]
catia=[true|false]
guestok=[true|false]
ro=access-list
rw=access-list
none=access-list
```

NFS および SMB 共有プロパティーの詳細については、share_nfs(1M) および share_smb(1M) を参照してください。

zfs set [-r] *property=value filesystem|volume|snapshot* ...
プロパティーを、各データセット用に指定した値に設定します。一部のプロパティーのみ、編集可能です。設定可能なプロパティーおよび有効な値については、「プロパティー」の節を参照してください。

`-r`
子データセットのサブツリー全体に対して、設定の有効な値を再帰的に適用します。有効な値は、プロパティーに応じて設定または継承できます。

`zfs set -c` share=*name*=*sharename filesystem*
ファイルシステム共有を削除します。`-c` オプションは、前述の `zfs set share` コマンドからこのサブコマンドを区別します。

`zfs share -a` | *dataset*
`sharenfs` または `sharesmb` プロパティーが設定されている場合に、共有プロパティー値に従って ZFS データセットの NFS または SMB 共有を公開します。

NFS または SMB プロトコルでデータセットを共有すると、データセットのデータがネットワークを介して使用できます。`sharenfs` または `sharesmb` プロパティーが設定されている ZFS データセットは、システムのブート時に自動的に共有されます。

`-a`
共有プロパティー値、および `sharenfs` プロパティーと `sharesmb` プロパティーの設定に従って、すべての ZFS データセットを共有します。

dataset
共有プロパティー値、および `sharenfs` プロパティーと `sharesmb` プロパティーの設定に従って、指定されたデータセットを共有します。

`zfs unshare -a` | *dataset*|*mountpoint*
`sharenfs` プロパティーまたは `sharesmb` プロパティーが設定されているすべての ZFS データセットを共有解除します。

`-a`
すべての共有されている ZFS データセットを共有解除します。これは、ブートプロセスの一部として自動的に呼び出されます。

dataset|*mountpoint*
指定されたデータセットを共有解除します。このコマンドには、システム上で共有される ZFS データセットのパスを指定することもできます。

使用例

例 1　ZFS ファイルシステムの NFS または SMB 共有を作成する

ZFS ファイルシステムの NFS または SMB 共有の作成プロセスは、次の 2 つの手順で構成されます。

1. `zfs set share` コマンドで NFS または SMB 共有プロパティーを識別して、共有を作成します。

   NFS 共有を作成するには:

```
# zfs create -o mountpoint=/data tank/data
# zfs set share=name=data,path=/data,prot=nfs tank/data
```

   SMB 共有を作成するには:

例 1　ZFS ファイルシステムの NFS または SMB 共有を作成する　(続き)

```
# zfs set share=name=fs,path=/fs,prot=smb,ro=@10-132-60-234 \
tank/fs
```

2. sharenfs プロパティーまたは sharesmb プロパティーを設定して共有を公開します。次のようにコマンドを使用します。

```
# zfs set sharenfs=on tank/data
# zfs set sharesmb=on tank/fs
```

例 2　NFS または SMB 共有情報を表示する

NFS または SMB 共有情報を表示します。

```
# zfs get share
NAME                      PROPERTY  VALUE  SOURCE
tank/data                 share     name=data,path=/data,prot=nfs  local
tank/fs                   share     name=fs,path=/fs,prot=smb,ro... local
```

例 3　ZFS ファイルシステムの共有プロパティーを追加または変更する

次のように、hostA に読み取り専用などの新しいオプションを追加して、共有を変更します。

```
# zfs set share=name=data,prot=nfs,ro=hostA tank/data
```

例 4　ZFS ファイルシステムの NFS または SMB 共有を削除する

zfs set -c コマンドを使用して共有を削除します。

```
# zfs set -c share=name=fs tank/fs
share 'fs' was removed.
```

終了ステータス

次の終了値が返されます。

0
　正常終了。

1
　エラーが発生した。

2
　無効なコマンド行オプションが指定された。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	確実

関連項目

chmod(1), chown(1), pktool(1), ssh(1), mount(1M), shadowd(1M), share(1M), share_nfs(1M), share_smb(1M), unshare(1M), zfs(1M), zfs_allow(1M), zfs_encrypt(1M), zonecfg(1M), zpool(1M), chmod(2), chown(2), stat(2), write(2), fsync(3C), dfstab(4), vfstab(4), attributes(5)

gzip(1) のマニュアルページを参照してください。これは SunOS マニュアルページコレクションには含まれていません。

その他の ZFS 機能の使用についての詳細は、『Oracle Solaris の管理: ZFS ファイルシステム』を参照してください。

名前

zic – time zone compiler

形式

```
zic [--version] [-s] [-v] [-d directory | -l localtime]
    [-p posixrules] [-y yearistype] [filename]...
```

機能説明

`zic` reads text from the file(s) named on the command line and creates the time conversion information files specified in this input. If a *filename* is '–', the standard input is read.

Input lines are made up of fields. Fields are separated by any number of white space characters. Leading and trailing white space on input lines is ignored. A pound sign (#) indicates a comment that extends to the end of the line. White space characters and pound signs can be enclosed within double quotes (" ") if they are to be used as part of a field. Any line that is blank (after comment stripping) is ignored. Non-blank lines are expected to be of one of three types: rule lines, zone lines, or link lines.

Rule

A rule line has the form:

For example:

```
Rule   NAME  FROM  TO  TYPE  IN   ON     AT   SAVE  LETTER/S
```

The fields that make up a rule line are:

```
Rule   USA   1969  1973   -  Apr lastSun 2:00  1:00   D
```

`NAME` Gives the (arbitrary) name of the set of rules this rule is part of.

`FROM` Gives the first year in which the rule applies. The word `minimum` (or an abbreviation) means the minimum year with a representable time value. The word `maximum` (or an abbreviation) means the maximum year with a representable time value.

`TO` Gives the final year in which the rule applies. In addition to `minimum` and `maximum` (as above), the word `only` (or an abbreviation) can be used to repeat the value of the `FROM` field.

`TYPE` Gives the type of year in which the rule applies. If `TYPE` is:

- '–' The rule applies in all years between `FROM` and `TO`, inclusive.
- `uspres` The rule applies in U.S. Presidential election years.
- `nonpres` The rule applies in years other than U.S. Presidential election years.
- `even` The rule applies to even-numbered years.
- `odd` The rule applies to odd-numbered years.

If `TYPE` is something else, then `zic` will attempt to execute the command

```
yearistype year type
```

to check the type of a year: an exit status of `0` means that the year is of the given type; an exit status of `1` means that the year is not of the given type. The `yearistype` command is not currently provided in the Solaris environment.

`IN` Names the month in which the rule takes effect. Month names can be abbreviated.

`ON` Gives the day on which the rule takes effect. Recognized forms include:

`5`	the fifth day of the month
`lastSun`	The last Sunday in the month
`lastMon`	The last Monday in the month
`Sun>=8`	First Sunday on or after the eighth
`Sun<=25`	Last Sunday on or before the 25th

Names of days of the week can be abbreviated or spelled out in full. Note: There cannot be spaces within the `ON` field.

`AT` Gives the time of day at which the rule takes effect. Recognized forms include:

`2`	Time in hours
`2:00`	Time in hours and minutes
`15:00`	24-hour format time (for times after noon)
`1:28:14`	Time in hours, minutes, and seconds, where hour 0 is midnight at the start of the day and hour 24 is midnight at the end of the day.

Any of these forms can be followed by the letter `w` if the given time is local “wall clock” time; `s` if the given time is local “standard” time; or `u` (or `g` or `z`) if the given time is universal time. In the absence of an indicator, wall clock time is assumed.

`SAVE` Gives the amount of time to be added to local standard time when the rule is in effect. This field has the same format as the `AT` field (without the `w` and `s` suffixes).

`LETTER/S` Gives the “variable part” (for example, the “S” or “D” in “EST” or “EDT” of time zone abbreviations to be used when this rule is in effect. If this field is '–', the variable part is null.

`Zone` A zone line has the form:

```
Zone  NAME                GMTOFF  RULES/SAVE   FORMAT  [UNTIL]
```

For example:

```
Zone Australia/SouthWest   9:30        -         CST    1992 Mar 15 12:00
                           8:30      Aus         CST
```

The fields that make up a zone line are:

NAME
: The name of the time zone. This is the name used in creating the time conversion information file for the zone.

GMTOFF
: The amount of time to add to UTC to get standard time in this zone. This field has the same format as the AT and SAVE fields of rule lines; begin the field with a minus sign to subtract time from UTC.

RULES/SAVE
: The name of the rule(s) that apply in the time zone or, alternately, an amount of time to add to local standard time. If this field is '–', then standard time always applies in the time zone.

FORMAT
: The format for time zone abbreviations in this time zone. The pair of characters %s is used to show where the "variable part" of the time zone abbreviation goes. Alternately, a slash (/) separates standard and daylight abbreviations.

UNTIL
: The time at which the UTC offset or the rule(s) change for a location. It is specified as a year, a month, a day, and a time of day. The time of day has the same format as the AT field of rule lines. If this is specified, the time zone information is generated from the given UTC offset and rule change until the time specified.

  The month, day, and time of day have the same format as the IN, ON, and AT columns of a rule; trailing columns can be omitted, and default to the earliest possible value for the missing columns.

  The next line must be a "continuation" line. This line has the same form as a zone line except that the string "Zone" and the name are omitted. The continuation line places information starting at the time specified as the UNTIL field in the previous line in the file used by the previous line. Continuation lines can contain an UNTIL field, just as zone lines do, indicating that the next line is a further continuation.

Link

A link line has the form:

```
Link   LINK-FROM   LINK-TO
```

For example:

```
Link   Europe/Istanbul Asia/Istanbul
```

The LINK-FROM field should appear as the NAME field in some zone line; the LINK-TO field is used as an alternate name for that zone.

Except for continuation lines, lines can appear in any order in the input.

オプション

`--`*version* Outputs version information and exits.

`-d` *directory* Creates time conversion information files in the directory *directory* rather than in the standard directory `/usr/share/lib/zoneinfo`.

The `-d` and `-l` options are mutually exclusive.

`-l` *localtime* Uses the given time zone as local time *localtime*. `zic` acts as if the file contained a link line of the form:

```
Link localtime localtime
```

The `-d` and `-l` options are mutually exclusive.

`-p` *posixrules* Uses the rules of the given time zone *posixrules* when handling POSIX-format time zone environment variables. `zic` acts as if the input contained a link line of the form:

```
Link    posixrules posixrules
```

This option is not used by `ctime(3C)` and `mktime(3C)` in the Solaris environment.

`-s` Limits time values stored in output files to values that are the same whether they are taken to be signed or unsigned. You can use this option to generate SVVS-compatible files.

This option is obsolete and may be removed in a future release.

`-v` Complains if a year that appears in a data file is outside the range of years representable by system time values (`0:00:00 a.m. UTC, January 1, 1970`, to `3:14:07 a.m. UTC, January 19, 2038`). This option also complains if a time of 24:00 (which cannot be handled by pre-1998 versions of `zic`) appears in the input.

`-y` *yearistype* Uses the given command *yearistype* rather than `yearistype` when checking year types (see `Rules` under `DESCRIPTION`).

オペランド

filename A file containing input lines that specify the time conversion information files to be created. If a *filename* is '–', the standard input is read.

ファイル

`/usr/share/lib/zoneinfo`	Standard directory used for created files
`/usr/share/lib/zoneinfo/src`	Directory containing source files

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed*

* The -s option is obsolete.

関連項目

time(1), zdump(1M), ctime(3C), mktime(3C), attributes(5)

注意事項

For areas with more than two types of local time, you might need to use local standard time in the AT field of the earliest transition time's rule to ensure that the earliest transition time recorded in the compiled file is correct.

If the current *timezone* file is edited and compiled using the "zic" command, the changes will only be reflected in any new processes that are running. The most accurate way to reflect the changes for the whole system would be a reboot.

名前

zoneadm – ゾーンの管理

形式

```
zoneadm -z zonename [-u uuid-match] subcommand
      [subcommand_options]

zoneadm [-R root] [-z zonename] [-u uuid-match] list
      [list_options]

zoneadm [-R root] -z zonename [-u uuid-match] mark incomplete
```

機能説明

zoneadm ユーティリティーは、システムゾーンを管理するために使用されます。ゾーンは、オペレーティングシステムによって実行時に管理されるアプリケーションコンテナです。

セキュリティー

ゾーン 0 以外のゾーンに実装されたプロセスは、すべての子プロセスを含めて、ゾーンを変更できません。

セキュリティー

単純な一覧表示機能とヘルプ機能以外は、大域システムゾーンで操作しているユーザーのみが zoneadm を使用でき、ルートの実効ユーザー ID で実行する必要があります。さらに、ユーザーは、特定のサブコマンドを実行することが承認されている必要があります。

zoneadm は、前にスラッシュ文字が付き、接尾辞として指定された *zonename* が任意で含まれる承認文字列をチェックします。省略すると、承認はすべてのゾーンと一致します。

情報を表示するだけのサブコマンド (help や list など) では、承認は必要ありません。ほかのすべてのサブコマンドでは、承認 solaris.zone.manage/*zonename* が必要となります。

ゾーン 0 以外のゾーンに実装されたプロセスは、すべての子プロセスを含めて、ゾーンを変更できません。

オプション

次のオプションがサポートされています。

-R *root*
: 代替ルート (ブート環境) を指定します。このオプションは、「list」および「mark」サブコマンドとの組み合わせでのみ使用できます。

-u *uuid-match*
: libuuid(3LIB) によって割り当てられる、ゾーンの一意識別子。このオプションが存在し、引数が空の文字列でない場合、UUID に一致するゾーンがあると -z オプションで指定されたゾーンの代わりに選択されます。

-z *zonename*
: ゾーンの文字列識別子。

サブコマンド

破壊的な動作や作業内容の消失を伴う可能性のあるサブコマンドには、強制的にその処理を実行するために -F フラグが用意されています。端末デバイスから入力しているときに、-F フラグを指定しないでそのようなコマンドを実行した場合は、フラグを指定するかどうかを確認されます。それ以外の状況で -F フラグを指定しないでそのようなコマンドを実行した場合、その操作は許可されず、診断メッセージが標準エラーに書き込まれます。ゾーンのインストールまたはアンインストールが中断した場合、ゾーンの状態は不完全なままになります。そのようなゾーンをリセットして構成済みの状態に戻すには、アンインストールを使用します。

次のサブコマンドがサポートされています。

attach [-u] [-F] [-n *path*] [*brand-specific options*]

attach サブコマンドは、あるシステムから切り離されたゾーンを取得し、そのゾーンを新しいシステムに接続します。したがって、「接続」を行う前に、detach サブコマンドを実行することをお勧めします (必須ではありません)。構成済み状態の新しいゾーンが得られたら、ゾーンを新しいゾーンとしてインストールする代わりに、attach サブコマンドを使用してゾーンのルートを設定します。

-F オプションを使えば、確認を行わずにゾーンを強制的に「インストール済み」の状態にすることができます。あるソースシステムからゾーンを適切にホストできないターゲットシステムへの移動を行なった場合にゾーンがサポート不可能な状態に陥る危険性があるため、このオプションは注意して使用してください。-n オプションを使用すると、attach サブコマンドの「予行演習」を実行できます。このオプションは、「detach -n」サブコマンドの出力を入力として使用し、ネットワークデバイスの非互換など、あらゆる競合問題を特定するのに便利です。また、ホストがゾーンをサポートできるかどうかを判定することもできます。パスには「-」を指定できますが、その場合、入力が標準入力から読み取られます。

ゾーンのブランドには、ゾーンがどのように接続されるかを管理する、追加のオプションが含まれる場合があります。ブランド固有の情報については、brands(5) を参照してください。

接続するゾーンは、事前に zonecfg (zonecfg(1M) を参照) コマンドを使って構成しておく必要があります。ただしそれは、「attach - n」を実行する場合には当てはまりません。

ゾーンを接続するには次のコマンドを使用します。

```
# zoneadm -z my-zone attach
```

ゾーンの接続と更新を行うには、次のコマンドを使用します。

```
# zoneadm -z my-zone attach -u
```

-n を指定しない場合 (上記の例)、このサブコマンドを使用する前にソースゾーンを停止する必要があります。

-n *path*
ゾーンのマニフェストを読み取り、実際に接続を実行することなく、ターゲットマシンがゾーンをホストできるように正しく構成されているかどうかを確認します。試行接続を行う前に、新規ホストで移行先システムのゾーンを構成する必要はありません。

-u
接続されたゾーンを更新します。

boot [-w|-W] [-- *boot_options*]
指定されたゾーンをブート (アクティブに) します。

boot サブコマンドには、次のような相互に排他的なオプションがあります。

-w
書き込み可能なルートでゾーンをブートします。これにより、ゾーンの構成で効率的に file-mac-profile 設定が上書きされます。このオプションは、このブートサイクルでのみ有効です。その後のリブートでは、file-mac-profile を含むゾーンが効率的に再度ブートされます。

-W
一時的な R/W モードでゾーンをブートします。ゾーンが自己アセンブリを完了すると、ゾーンは読み取り専用モードでリブートします。 非読み取り専用のルートゾーンでは無効です。

次の *boot_options* がサポートされています。

-i *altinit*
代替実行可能ファイルを原始プロセスとして選択します。 *altinit* は実行可能ファイルへの有効なパスです。デフォルトの原始プロセスは init(1M) です。

-m *smf_options*
smf_options には、サービス管理機能のブート動作を制御する次の 2 つのカテゴリのオプションが含まれています。 復旧オプションとメッセージオプションです。

メッセージオプションは、ブート中に smf(5) が表示するメッセージの種類と量を決定します。サービスオプションは、システムのブートに使用されるサービスを決定します。-m サブオプションの一覧については、kernel(1M) を参照してください。

-s
マイルストン svc:/milestone/single-user:default に対してのみブートします。このマイルストーンは、init のレベル s と同等です。svc.startd(1M) および init(1M) を参照してください。

`clone` [`-m` *copy*] [`-s` *zfs_snapshot*] [*brand-specific options*] *source_zone*

既存のインストール済みゾーンをコピーすることでゾーンのインストールを行います。このサブコマンドは、ゾーンをインストールするための代替手段となります。

`-m` *copy*

「ZFS クローン」が可能な場合でも、このクローンを強制的にコピーにします。

`-s` *zfs_snapshot*

クローンのソースとして使用する ZFS スナップショットの名前を指定します。*snapshot* は、以前の「`zoneadm` clone」インストールから取得したソースゾーンの *snapshot* でなければなりません。

ソースゾーンを停止しないと、このサブコマンドを使用できません。

`detach` [`-F` | `-n`]

指定されたゾーンを切り離します。ゾーンの切り離しは、あるシステムから別のシステムへゾーンを移動する際の、最初のステップです。ゾーンを切り離し、*zonepath* ディレクトリを新しいホストに移動したあと、ゾーンを新しいホストに接続する、というのが、完全なゾーン移行手順となります。ゾーンが切り離されると、そのゾーンは構成済みの状態になります。切り離された構成済みゾーンをインストールまたはクローンしようとするとエラーメッセージが表示され、その `install` または `clone` サブコマンドの処理を続行できなくなります。`-n` オプションを使用すると、`detach` サブコマンドの「予行演習」を実行できます。この場合、「`attach -n`」サブコマンドの実行に必要な情報が生成されます。このサブコマンドは、ネットワークデバイスの非互換やホストのゾーン未サポートなど、あらゆる競合問題を特定するのに便利です。この情報は標準出力に送られますが、ファイルに保存したり、「`attach -n`」サブコマンドにパイプしたりすることもできます。`-F` オプションを使用して、既存の *zonepath* の検証チェックを実行せずにゾーンを強制的に切り離すことができます。

ゾーンを切り離すには次のコマンドを使用します。

```
# zoneadm -z my-zone detach
```

`-n` オプションを指定しないかぎり、このサブコマンドを使用する前にソースゾーンを停止する必要があります。

`-F`

ゾーンの記憶領域の検証チェックを実行せずにゾーンを強制的に切り離します。このオプションは通常、*zonepath* の記憶領域にこのホストがアクセスできなくなっているときに使用されます。このようなシナリオは通常、手動で、またはクラスタの一部として、ゾーンの記憶領域が代替ホストにフェイルオーバーされている場合に発生します。

`-n`

実際にゾーンを切り離すことなく、実行中のゾーンでゾーンのマニフェストを生成します。移行元のシステムのゾーンの状態は変わりません。ゾーンの

マニフェストは標準出力に送信されます。大域管理者は、この出力をファイルに送ったり、移行先ホストですぐに検証されるように遠隔コマンドにパイプしたりできます。

`halt`
指定されたゾーンを停止します。`halt` を指定した場合、そのゾーンのシャットダウンスクリプトは実行されません。また、ゾーンの実行時リソースを削除します。

次の `shutdown` サブコマンドも参照してください。

`help` [*subcommand*]
一般ヘルプを表示します。*subcommand* を指定した場合は、*subcommand* に関するヘルプが表示されます。

`install` [*brand-specific options*]
指定されたゾーンをシステムにインストールします。このサブコマンドが実行される前に、自動的にゾーンの確認が行われます。この確認手順が失敗した場合は、インストールが拒否されます。`verify` サブコマンドを参照してください。

ゾーンのブランドには、ソフトウェアがゾーンにどのようにインストールされるかを管理する、追加のオプションが含まれる場合があります。ブランド固有の情報については、`brands(5)` を参照してください。

`list` [*list_options*]
現在のゾーンの名前、またはゾーンが表示されるように指定されている場合はそのゾーンの名前を表示します。

デフォルトでは、実行中のすべてのゾーンの一覧を表示します。このサブコマンドを `zoneadm -z` *zonename* オプションと一緒に使用した場合は、指定したゾーンの状態に関係なく、そのゾーンだけが表示されます。この場合、`-i` および `-c` オプションは許可されません。

`-i` オプションと `-c` オプションのどちらも指定されなかった場合、実行中のすべてのゾーンの一覧を表示します。

次の *list_options* がサポートされています。

`-c`
構成済みのすべてのゾーンを表示します。このオプションは、`-i` オプションよりも優先されます。

`-i`
インストール済みのすべてのゾーンを表示します。

`-p`
機械可読な出力を要求します。出力形式は行のリストです。1 つのゾーンは 1 行に出力され、各フィールドはコロンで区切られます。これらのフィールドは次のとおりです。

```
zoneid:zonename:state:zonepath:uuid:brand:ip-type:\
r/w:file-mac-profile
```

zonepath にコロンが含まれている場合、そのコロンはバックスラッシュ (“\: ”) でエスケープできます。これは、環境変数 IFS を持つシェルの read(1) 関数を使用することで解析できます。*uuid* 値は、ゾーンのインストール時に libuuid(3LIB) によって割り当てられ、代替ブート環境に同じゾーンが存在する (またはリネームされた) 場合にゾーンを識別するのに役立ちます。「zoneadm list -p」コマンドの出力を解析するすべてのソフトウェアは、将来追加される可能性のあるすべてのフィールドを扱うことができる必要があります。

-v と -p オプションを同時に指定することはできません。-v と -p のどちらのオプションも使用しない場合は、ゾーン名だけが表示されます。

-v

ゾーン名、ID、現在の状態、ルートディレクトリ、ブランドタイプ、IP タイプ、およびオプションなどの冗長情報を表示します。

-v と -p オプションを同時に指定することはできません。-v と -p のどちらのオプションも使用しない場合は、ゾーン名だけが表示されます。

mark incomplete

インストールされているゾーンの状態を「incomplete」に変更します。このコマンドは、システムにおける管理上の変更によって、ゾーンが使用できない、または一貫性のない状態になった場合に役立つ可能性があります。この変更は (ゾーンをアンインストールしないかぎり) 取り消せません。

move *new_zonepath*

zonepath を *new_zonepath* に移動します。ゾーンを停止しないと、このサブコマンドを使用できません。*new_zonepath* はローカルファイルシステムでなければなりません。また、*zonepath* の通常の制約が適用されます。

ready

アプリケーションを実行するためにゾーンを準備します。このゾーンでユーザープロセスを開始するためのサブコマンドではありません。

reboot [-- *boot_options*]

ゾーンを再起動します。これは、halt boot シーケンスと同等です (shutdown スクリプトは実行されません)。

サポートされているブートオプションについては、boot サブコマンドを参照してください。

shutdown [-r [-- *boot_options*]]

ゾーンを正常にシャットダウンします (ゾーンで /usr/sbin/init 0 を実行することと同等です)。 shutdown サブコマンドは、ゾーンが正常にシャットダウンされるまで待機します。シャットダウンプロセスに長時間かかる場合は、zoneadm halt を使用すればゾーンを強制的に停止できます。

-r が指定されている場合は、ゾーンをリブートします。サポートされているブートオプションについては、boot サブコマンドを参照してください。

uninstall [-F]
指定されたゾーンをシステムからアンインストールします。このサブコマンドを使用するときは、注意が必要です。指定されたゾーンの *zonepath* にあるすべてのファイルが削除されます。-F フラグを使用すれば、強制的に処理を実行できます。

verify
指定されたゾーンの構成を確認して、このマシンに安全にインストールできることを確認します。resource/property 形式による確認を次に示します。

zonepath
zonepath とその親ディレクトリが存在し、それらが適切なモードの root によって所有されていることを確認します。適切なモードとは、*zonepath* が 700 である、その親がグループ書き込み可やだれでも書き込み可でない、などの状態であることを指します。*zonepath* が NFS マウントを越えていないことを確認します。*zonepath* に「root」という名前のサブディレクトリが存在しないことを確認します。

zonepath が存在しなくても、verify が失敗することはありません。次回のインストールのときに適切なアクセス権で作成されることが警告されるだけです。それ以降の verify で問題が発生した場合には、失敗する可能性があります。

zonepath はシンボリックリンクであってはいけません。

fs
fs リソースの *type* 値を確認します。値が proc、mntfs、autofs、または nfs の場合、あるいはファイルシステムに関連付けられているマウントバイナリが /usr/lib/fs/<*fstype*>/mount に存在しない場合には、エラーを報告します。

directory が相対パスの場合は、エラーになります。

また、raw に指定されているパスが相対パスの場合、または指定されたファイルシステムタイプの fsck バイナリが /usr/lib/fs/<*fstype*>/fsck に存在しない場合は、エラーになります。対応する fsck バイナリは存在していても、raw パスが指定されていない場合には、エラーになります。

net
すべての物理ネットワークインタフェースが存在することを確認します。すべてのネットワークアドレスリソースは、次のいずれかになります。

- 有効な IPv4 アドレス。後続の「/」とプレフィックス長は任意です。
- 有効な IPv6 アドレス。後続の「/」とプレフィックス長は必須です。
- IPv4 アドレスに解決されるホスト名。

IPv6 アドレスに解決されるホスト名はサポートされていません。

物理インタフェース名はネットワークインタフェース名です。

ゾーンは排他的 IP か共有 IP のいずれかに構成できます。共有 IP ゾーンでは、物理プロパティーとアドレスプロパティーの両方を設定する必要があります。排他的 IP ゾーンでは、物理プロパティーを設定する必要があり、アドレスプロパティーは設定できません。

`anet`
VNIC が自動的に作成される下位リンクが存在することを確認します。

`rctl`
定義されているリソース制御値が現在のマシン上で有効であることも確認します。つまり、特権レベルが `privileged` であること、制限値が現在定義されているシステム値より低いこと、および定義されているアクションがリソース制御で有効になっていることを確認します。

使用例

例 1 -m オプションを使用する

次のコマンドは、`-m` オプションの使用法を示しています。

```
# zoneadm boot -- -m verbose
```

例 2 -i オプションを使用する

次のコマンドは、`-i` オプションの使用法を示しています。

```
# zoneadm boot -- -i /usr/sbin/init
```

例 3 -s オプションを使用する

次のコマンドは、`-s` オプションの使用法を示しています。

```
# zoneadm boot -- -s
```

終了ステータス

次の終了値が返されます。

`0`
正常終了。

`1`
エラーが発生しました。

`2`
無効な使用法。

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	確実

関連項目

read(1), svcs(1), zlogin(1), zonename(1), init(1M), kernel(1M), svcadm(1M), svc.startd(1M), svc.startd(1M), zonecfg(1M), libuuid(3LIB), attributes(5), brands(5), mwac(5), smf(5), zones(5)

注意事項

zones(5) サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/zones:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスステータスを照会するには、svcs(1) コマンドを使用します。

読み取り専用ゾーンのインストールまたはアップグレード後にはじめてブートしたとき、または -w/-W を使ってゾーンをブートしたときに、書き込み専用保護は無効化されます。 ゾーンが別の方法で保護されるように配慮する必要があります。

名前

zoneadmd – ゾーン管理デーモン

形式

/usr/lib/zones/zoneadmd

機能説明

zoneadmd は、システムが特定のゾーンを管理する必要があるときに起動されるシステムデーモンです。zoneadmd デーモンの各インスタンスが特定のゾーンを管理するため、複数の zoneadmd デーモンが実行されることがあります。

このデーモンは、ゾーン管理ソフトウェアにより自動的に起動されるので、直接起動しないようにしてください。デーモンは使用されなくなると自動的にシャットダウンします。これは、プログラミングインタフェースの構成要素ではありませんが、非公開インタフェースに分類されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	非公開

関連項目

svcs(1), zlogin(1), svcadm(1M), zoneadm(1M), attributes(5), smf(5), zones(5)

注意事項

zones(5) サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/zones:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスステータスを照会するには、svcs(1) コマンドを使用します。

名前

zonecfg – ゾーン構成の設定

形式

`zonecfg -z` *zonename*

`zonecfg -z` *zonename subcommand*

`zonecfg -z` *zonename* `-f` *command_file*

`zonecfg help`

機能説明

`zonecfg` ユーティリティーは、ゾーンの構成を作成、変更、および一覧表示します。作成機能と変更機能は承認されたユーザーだけが使用でき、ルートの実効ユーザー ID でプロセスを実行する必要があります。それ以外の場合には、読み取り専用モードで実行されます。

ゾーン構成は、いくつかのリソースおよびプロパティーで構成されます。

ユーザーインタフェースを簡素化するために、`zonecfg` では有効範囲の概念が使用されます。デフォルトの有効範囲は大域です。

次の形式の `zonecfg` コマンドは、対話形式で使用されます。

`zonecfg -z` *zonename subcommand*

`zonecfg` 経由で変更されたパラメータは、稼働中のゾーンには影響しません。変更を適用するには、ゾーンをリブートする必要があります。

ゾーンの作成と変更のほか、`zonecfg` ユーティリティーは大域ゾーンのリソース管理設定を永続的に指定するためにも使用できます。

以降の文では、「rctl」を「resource control」(リソース制御) の略語として使用します。`resource_controls(5)` を参照してください。

すべてのゾーンに、それぞれ関連するブランドが構成されます。ブランドによって、ゾーン内で使用されるユーザーレベルの環境のみでなく、ゾーンがインストール、ブート、またはシャットダウンされるときのゾーンのさまざまな動作も決まります。ゾーンのインストールが完了したあとは、そのブランドを変更することはできません。デフォルトのブランドは、大域ゾーンにインストールされているディストリビューションによって決まります。ブランドの中には、`zonecfg` のすべてのプロパティーおよびリソースをサポートしないものがあります。各ブランドについての詳細は、ブランド固有のマニュアルページを参照してください。ブランドの概要については、`brands(5)` のマニュアルページを参照してください。

リソース

次のリソースタイプがサポートされています。

`attr`
: 汎用属性。

`capped-cpu`
: CPU 使用率の制限。

`capped-memory`
物理メモリー、スワップメモリー、およびロックされたメモリーの制限。

`dataset`
ZFS データセット。

`dedicated-cpu`
稼働している間、このゾーン専用に割り当てられるシステムプロセッサのサブセット。

`device`
デバイス。

`fs`
ファイルシステム。

`net`
ネットワークインタフェース。

`anet`
自動ネットワークインタフェース。

`admin`
委任された管理者。

`rctl`
リソース制御。

疎ルートおよび完全ルート非大域ゾーン

Solaris の以前のリリースで使用されていた疎ルートゾーンの概念。この機能は、SVr4 パッケージングシステムと密接に関連しており、ディスク容量の節約と管理作業の軽減を目的としていました。

新しいパッケージングシステムである IPS では、ゾーンにインストールするパッケージをより柔軟に選択できるようになっています。したがって、ファイルシステムテクノロジが進歩したこと (特に ZFS 複製解除) も考慮すれば、疎ルートゾーンを削除することは非常に実用的です。IPS パッケージングとファイルシステムの進歩が組み合わされた結果、すべてのゾーンに疎ルートゾーンのメリットがもたらされています。

プロパティー

各リソースタイプには、1 つまたは複数のプロパティーが割り当てられます。また、いくつかの大域プロパティー (特定のリソースのプロパティーではなく、構成全体のプロパティー) も割り当てられます。

次のプロパティーがサポートされています。

(大域)
`zonename`

(大域)
`zonepath`

(大域)
`autoboot`

(大域)
`bootargs`

(大域)
`pool`

(大域)
`limitpriv`

(大域)
`brand`

(大域)
`cpu-shares`

(大域)
`hostid`

(大域)
`max-lwps`

(大域)
`max-msg-ids`

(大域)
`max-processes`

(大域)
`max-sem-ids`

(大域)
`max-shm-ids`

(大域)
`max-shm-memory`

(大域)
`scheduling-class`

(大域)
`fs-allowed`

(大域)
file-mac-profile

`fs`
`dir`、`special`、`raw`、`type`、`options`

`net`
`address`、`allowed-address`、`configure-allowed-address`、`physical`、`defrouter`

anet
  linkname、lower-link、allowed-address、configure-allowed-address、defrouter、mac-

device
  match

rctl
  name、value

attr
  name、type、value

dataset
  name、alias

dedicated-cpu
  ncpus、importance

capped-memory
  physical、swap、locked

capped-cpu
  ncpus

admin
  user、auths

これらのプロパティー名のプロパティー値は、単純値、複合値、またはリスト値で指定します。許可されるタイプはプロパティーごとに異なります。単純値は文字列で、引用符で囲むこともできます。複合値は次の構文で指定します。

(<*name*>=<*value*>,<*name*>=<*value*>,...)

ここで、各 <*value*> は単純値であり、<*name*> 文字列は与えられたプロパティー内で一意になります。リストは次の構文で指定します。

[<*value*>,...]

ここで、<*value*> は、単純値、複合値のいずれかです。値 (単純値または複合値) が 1 つだけのリストは、リスト構文を使わずにその値を指定することと等価です。つまり、「foo」は「[foo]」と等価です。リストは空でもかまいません (「[]」と表記)。

プロパティー値の解釈において、zonecfg は fnmatch(5) で指定されている正規表現に対応しています。「使用例」を参照してください。

次にプロパティータイプについて説明します。

global: zonename
  ゾーンの名前。

global: zonepath
  ゾーンのファイルシステムのパス。

global: autoboot
システムがブートするときに、ゾーンを自動的にブートするかどうかを指定するブール値。ゾーンサービスが無効になっている場合は、このプロパティーの設定に関係なく、ゾーンは自動的にブートされません。ゾーンサービスを有効にするには、次のように svcadm コマンドを使用します。

```
# svcadm enable svc:/system/zones:default
```

ゾーンサービスを無効にする場合は、enable を disable に置き換えます。svcadm(1M) を参照してください。

global: bootargs
ゾーンのブート時に渡される引数 (オプション)。ただし、zoneadm boot コマンドにオプションが指定された場合は、そちらが優先されます。有効な引数は、zoneadm(1M) で説明されています。

global: pool
ブート時にこのゾーンをバインドするリソースプールの名前。このプロパティーには dedicated-cpu リソースとの互換性はありません。

global: limitpriv
このゾーン内のいずれかのプロセスが入手できる特権のセットの最大数。プロパティーは、priv_str_to_set(3C) で説明されているコンマ区切りの特権セットの仕様で構成するようにしてください。名前の前にダッシュ (-) または感嘆符 (!) を付けると、結果として得られるセットから特権を除外できます。このコンテキストでは、特別な特権文字列「zone」はサポートされていません。特別な文字列「default」がプロパティーで最初のトークンとして出現する場合、zones(5) で説明されている、リソースとセキュリティーの分離を保持する安全な特権のセットに展開します。プロパティーが不足しているか空である場合、この同じ安全な特権のセットと同等になります。

システム管理者は、ゾーンの構成を行う際に細心の注意を払う必要があります。一部の特権は、ゾーンのブートに必須のためこのメカニズムを介して除外できません。さらに、ゾーン内のプロセスがほかのゾーンのプロセスに過度に影響を与えてしまうため、特定の特権はゾーンに対して付与されていません。zoneadm(1M) は、ゾーンを「ブート」または「準備」しようとする際、無効な特権がゾーンに追加されるまたはゾーンから削除される場合を示しています。

特権の説明は、privileges(5) を参照してください。コマンド「ppriv -l」(ppriv(1) を参照) は、すべての Solaris 特権の一覧を生成します。ppriv で表示されるように特権を指定できます。privileges(5) では、特権がPRIV_*privilege_name* の形式で一覧表示されています。たとえば、このプロパティーで特権 *sys_time* を指定した場合、これは privileges(5) では PRIV_SYS_TIME として表示されています。

global: brand
ゾーンのブランドタイプ。

global: ip-type
ゾーンは独自の排他的な IP のインスタンスを持つ (デフォルト) か、大域ゾーンと IP インスタンスを共有することができます。デフォルトゾーンテンプレート SYSdefault では、ip-type は exclusive に設定されます。追加で提供される SYSdefault-shared-ip テンプレートでは、ip-type は shared に設定されます。

このプロパティーは exclusive および shared の値を取ります。

global: hostid
ゾーンは、システムを統合しやすくするために、32 ビットのホスト識別子をエミュレートできます。ゾーンの hostid プロパティーはデフォルトで空になっており、ゾーンがホスト識別子をエミュレートしないことを意味しています。ゾーンのホスト識別子には、0 から FFFFFFFE の 16 進値を指定してください。接頭辞 0x または 0X は省略可能です。大文字と小文字の両方の 16 進数を使用できます。

global: fs-allowed
ゾーン内にマウントできる追加ファイルシステムのコンマ区切りリスト (たとえば、ufs,pcfs)。デフォルトでは、hsfs(7FS) とネットワークファイルシステムだけがマウントできます。

このプロパティーは、add fs または add dataset によってゾーンにマウントされたファイルシステムには適用されません。

注意 – デフォルト以外のファイルシステムマウントを許可すると、ゾーン管理者が偽のファイルシステムイメージでシステムに損害を与えることが可能になるため、このような許可はサポートされていません。

global: file-mac-profile
読み取り専用ポリシーから除外されるファイルシステムの部分、つまりゾーンが書き込むことができるファイルシステムの部分を定義します。

このプロパティーでは現在、none、strict、fixed-configuration、および flexible-configuration の 4 つの値がサポートされています。

none を指定すると、このゾーンは通常の r/w ゾーンと全く同じになります。strict を指定すると、読み取り専用ポリシーに対する例外を許可しません。fixed-configuration を指定すると、構成ファイルが格納されているディレクトリを除いて、/var の内部とそれ以下にあるファイルへの書き込みが許可されます。

```
/var/ld
/var/lib/postrun
/var/pkg
/var/spool/cron,
/var/spool/postrun
/var/svc/manifest
/var/svc/profiles
```

flexible-configuration は fixed-configuration と同じですが、/etc 内のファイルへの書き込みも追加で許可されます。

fs: dir、special、raw、type、options
ファイルシステムをマウントする方法や場所などを決めるために必要な値。mount(1M)、mount(2)、fsck(1M)、および vfstab(4) を参照してください。

net: address、allowed-address、configure-allowed-address、physical、defrouter
net リソースは、ゾーンに対する物理ネットワークリソースの割り当てを表します。割り当ての前に、リソースは大域ゾーンに存在する必要があります。

ネットワークインタフェースのネットワークアドレスと物理インタフェース名。ネットワークアドレスは、次のいずれかになります。

- 有効な IPv4 アドレス。後続の「/」とプレフィックス長は任意です。
- 有効な IPv6 アドレス。後続の「/」とプレフィックス長は必須です。
- IPv4 アドレスに解決されるホスト名。

IPv6 アドレスに解決されるホスト名はサポートされていません。

物理インタフェース名はネットワークインタフェース名です。

省略可能なデフォルトルートの値は、ネットワークアドレスと同様に指定しますが、後ろに / (スラッシュ) とネットワーク接頭辞長を決して付けないでください。defrouter 機能を正しく使用できるようにするには、このプロパティーを使用するゾーンを、大域ゾーンが常駐するサブネットとは異なるサブネット上に配置してください。また、異なる defrouter 設定を使用する各ゾーン (あるいはゾーンセット) を異なるサブネット上に配置してください。

ゾーンは排他的 IP か共有 IP のいずれかに構成できます。共有 IP ゾーンでは、物理プロパティーとアドレスプロパティーの両方を設定する必要があります。デフォルトルートの設定は省略可能です。物理プロパティーで指定したインタフェースは、非大域ゾーンをブートする前に大域ゾーンに plumb する必要があります。ただし、そのインタフェースが大域ゾーンで使用されない場合は、そのインタフェースを大域ゾーンで down に構成するようにしてください。そのインタフェースのデフォルトルートがここで指定されるはずです。allowed-address プロパティーは共有 IP ゾーン用には設定できません。

排他的 IP ゾーンでは、physical プロパティーを設定する必要があり、address プロパティーは設定してはいけません。また、排他的 IP ゾーンが使用できる IP アドレスのセットは、allowed-address プロパティーを指定して制限できます。allowed-address が指定されていない場合、排他的 IP ゾーンは、関連付けられた physical インタフェース上の任意の IP アドレスをネットリソースに使用できます。そうではなく、allowed-address が指定されている場合、排他的 IP ゾーンは、allowed-address リストにない IP アドレスを physical アドレスに使用できません。configure-allowed-address が true に設定されている場合、allowed-address は、インストール後の初期ブート時にインタフェースに永続的に構成されます。これが false に設定されている場合、allowed-address は

ゾーンのブート時に構成されません。デフォルトでは、allowed-address が指定されると configure-allowed-address は true に設定されます。さらに、allowed-address リストが入力されている場合は、defrouter プロパティーも必要に応じて指定できます。ただし、defrouter 値が指定されて configure-allowed-address が false に設定された場合、defrouter 値は無視されて、該当する警告メッセージが表示されます。physical プロパティーに指定されたインタフェースを大域ゾーンで使用してはいけません。allowed-address とデフォルトルーターが zonecfg によって指定された場合、これらは、非大域排他的 IP ゾーンで ipadm(1M) によってインタフェースが有効にされているとき (通常はゾーンの起動中) に、そのインタフェースに適用されます。非大域排他的 IP ゾーンは、ほかのどのアドレスをそのインタフェースに適用することもできなくなります。また、指定された IP バージョンの別のソースアドレスを持つパケットを送信することもできなくなります。zonecfg によって設定されたデフォルトのルーターは、-p フラグと route(1M) を使用して非大域排他的 IP ゾーン内から恒久的に削除することはできません。

複数の排他的 IP ゾーン間で 1 つのデータリンクを共有することはできません。

anet:

linkname、lower-link、allowed-address、configure-allowed-address、defrouter、mac-address、n

anet リソースは、排他的 IP ゾーンについてのネットワークリソースの自動作成を表します。zonecfg がデフォルトの SYSdefault テンプレートを使用してゾーンを作成すると、次のプロパティーを持つ anet リソースがゾーン構成に自動的に入ります。

```
linkname=net0
lower-link=auto
mac-address=random
link-protection=mac-nospoof
```

そのようなゾーンがブートすると、lower-link (次を参照) が自動的に決定され、そのリンク上にゾーンの一時的な VNIC が自動的に作成されます。ゾーンが停止すると、lower-link および VNIC は削除されます。

サポートされるプロパティーについては、下記を参照してください。これらのプロパティーはすべてオプションです。大域ゾーンだけが、自動的に作成された VNIC またはそのプロパティーを変更できます。zonecfg に設定されたプロパティーを VNIC の作成時に VNIC に割り当てることができない場合、ゾーンはブートに失敗します。

linkname

自動作成される VNIC の名前を指定します。デフォルトでは、このプロパティーは、net*N* という形式の (ゾーンの) 最初に利用可能な名前に自動的に設定されます。ここで、*N* は負でない整数です。たとえば、net0、net1 のようになります。info サブコマンドによって、自動的に選択される linkname が表示されます。

大域ゾーンを含む複数のゾーンが、同じ名前のリンクを同時に持つことができます。

`lower-link`
VNIC が作成されるリンクを指定します。このプロパティーのデフォルト値は `auto` です。管理者は、`anet` リソースを追加するとき、値を明示的に指定できます。リンクには、`dladm create-vnic` の `-l` オプションの引数として受け入れられるすべてのリンクが使用できます (`dladm(1M)` を参照してください)。このプロパティーが `linkname` (`auto` 以外) に設定され、そのリンクが存在しない場合、ゾーンはブートに失敗します。`auto` に設定されると、`zoneadmd(1M)` デーモンは、ゾーンがブートするたびに、VNIC が作成されるリンクを自動的に選択します。リンクは次の発見的方法を使用して選択されます。

1. リンク状態が `up` の最初の物理データリンク。
2. `up` のものがない場合、`net0` という名前のデータリンクが使用されます。

前述の両方が満たされない場合、ゾーンはブートに失敗します。

`allowed-address`
`net` リソースの排他的 IP ゾーンについての `allowed-address` プロパティーの説明を参照してください。

`configure-allowed-address`
`net` リソースの排他的 IP ゾーンについての `configure-allowed-address` プロパティーの説明を参照してください。

`defrouter`
`net` リソースの排他的 IP ゾーンについての `defrouter` プロパティーの説明を参照してください。

`mac-address`
指定された値またはキーワードに基づいて VNIC の MAC アドレスを設定します。値がキーワードでない場合は、ユニキャスト MAC アドレスとして解釈されます。サポートされるキーワードは次のとおりです。

- `factory`: ファクトリ MAC アドレスを VNIC に割り当てます。ファクトリ MAC アドレスが要求された場合、`mac-slot` プロパティーを使用して MAC アドレススロット識別子を指定できます。そうでない場合、次に使用可能なファクトリ MAC アドレスが使用されます。
- `random`: ランダムな MAC アドレスを VNIC に割り当てます。`mac-prefix` プロパティーを使用して接頭辞を指定します。そうでない場合、ローカルビットが設定された有効な IEEE OUI で構成されたデフォルトの接頭辞が使用されます。これがデフォルト値です。
- `auto`: 最初にファクトリ MAC アドレスの使用を試みます。どれも使用できない場合は、ランダムな MAC アドレスを割り当てます。

ランダムな MAC アドレスが選択された場合、生成されたアドレスは、ゾーンのブート前後およびゾーンの切り離しと接続の前後で保持されます。これに

より、ゾーンは安定したクライアント ID を維持することによって DHCP リースを保持できるほか、安定した MAC アドレスを持つことによるほかの利点を活用することができます。

mac-prefix
ランダムな MAC アドレスが要求された場合の MAC アドレス接頭辞を指定します。それ以外の場合、このプロパティーは無視されます。

mac-slot
ファクトリ MAC アドレスが要求された場合の MAC アドレススロット識別子を指定します。それ以外の場合、このプロパティーは無視されます。

mtu
VNIC の最大転送単位 (バイト)。dladm(1M) の mtu プロパティーを参照してください。

maxbw
VNIC の全二重帯域幅を指定します。dladm(1M) の maxbw プロパティーを参照してください。デフォルトでは、VNIC は lower-link に設定された maxbw を使用し、何も設定されない場合は帯域幅制限はありません。

priority
VNIC の相対的な優先順位を指定します。サポートされる値とデフォルトについては、dladm(1M) の priority プロパティーを参照してください。

vlan-id
この VNIC の VLAN タグを使用可能にして、VLAN タグの ID を指定します。デフォルト値は存在しません。つまり、このプロパティーを設定しないと、VNIC はどの VLAN にも参加しません。

rxfanout
受信側ファンアウトスレッドの数を指定します。dladm(1M) の説明を参照してください。

rxrings
VNIC の受信リングを指定します。サポートされる値とデフォルトについては、dladm(1M) の rxrings プロパティーを参照してください。

txrings
VNIC の送信リングを指定します。サポートされる値とデフォルトについては、dladm(1M) の txrings プロパティーを参照してください。

link-protection
コンマ区切り値を使用して、1 つ以上のリンク保護のタイプを使用可能にします。サポートされる値については、dladm(1M) の protection プロパティーを参照してください。このデフォルト値は mac-nospoof です。

ip-nospoof をこのプロパティーに追加しても、allowed-address も設定しないかぎり効果はありません。allowed-address を設定すると、ip-nospoof が link-protection のセットに暗黙に追加され、allowed-address をクリアするとこれが削除されます。

allowed-dhcp-cids
このプロパティーを設定すると、VNIC の dhcp-nospoof が有効になります。詳細は、dladm(1M) を参照してください。

device: match
照合するデバイス名。これは、照合するパターンにすることも絶対パス名にすることもできます。デバイスリソースと別名を付けたデータセットは、/dev/zvol 内で名前空間の競合が発生する可能性があります。dev(7FS) を参照してください。

注 – 一般に、ゾーンにデバイスを追加すると、システムのセキュリティーが損なわれる可能性があります。「注意事項」を参照してください。

rctl: name、value
リソース制御の名前、特権、制限、アクション。prctl(1) および rctladm(1M) を参照してください。rctl の値の望ましい設定方法は、特定の rctl に関連する大域的なプロパティー名を使用する方法です。

attr: name、type、value
汎用属性の名前、型、および値。type は、int、uint、boolean、string のいずれかでなければなりません。値には、その型の値を指定する必要があります。uint は符号なしの (負でない) 整数です。

dataset: name、alias
ゾーン内からアクセスする ZFS データセットの名前。zfs(1M) を参照してください。各データセットがゾーン内の仮想 ZFS プールとして表示されるように、データセットに別名が付けられます。別名はこの仮想プールの名前です。ZFS プール名に適用される名前制限と、その結果、データセットの別名の値にも適用される名前制限については、zpool(1M) を参照してください。別名 rpool はゾーンの rpool データセットから予約されています。別名を付けたデータセットとデバイスリソースは、/dev/zvol 内で名前空間の競合が発生する可能性があります。dev(7FS) を参照してください。

global: cpu-shares
このゾーンに割り当てられている公平配分スケジューラ (FSS) の配分数。このプロパティーには dedicated-cpu リソースとの互換性はありません。このプロパティーは、zone.cpu-shares rctl を設定するための望ましい方法です。

global: max-lwps
このゾーンの、同時に使用できる LWP の最大数。このプロパティーは、zone.max-lwps rctl を設定するための望ましい方法です。

global: max-msg-ids
このゾーンに許容されるメッセージキュー ID の最大数。このプロパティーは、zone.max-msg-ids rctl を設定するための望ましい方法です。

global: max-processes
このゾーンの、同時に使用できるプロセステーブルスロットの最大数。このプロパティーは、zone.max-processes rctl を設定するための望ましい方法です。このプロパティーを設定すると、max-lwps プロパティーが明示的に設定されていないかぎり、max-lwps プロパティーの値がプロセススロット数の 10 倍の数に暗黙的に設定されます。

global: max-sem-ids
このゾーンに許容されるセマフォー ID の最大数。このプロパティーは、zone.max-sem-ids rctl を設定するための望ましい方法です。

global: max-shm-ids
このゾーンに許容される共有メモリー ID の最大数。このプロパティーは、zone.max-shm-ids rctl を設定するための望ましい方法です。

global: max-shm-memory
このゾーンに許容される共有メモリーの最大容量。このプロパティーは、zone.max-shm-memory rctl を設定するための望ましい方法です。この数値には、K、M、G、T の単位を適用できます。たとえば、1M は 1 メガバイトです。

global: scheduling-class
ゾーンで実行されているプロセスに使用するスケジューリングクラスを指定します。このプロパティーを指定しないと、スケジューリングクラスは次のように確立されます。

- cpu-shares プロパティーまたは同等の rctl が設定されている場合、スケジューリングクラス FSS が使用されます。
- cpu-shares も同等の rctl も設定されず、ゾーンのプールプロパティーがデフォルトのスケジューリングクラスを持つプールを参照する場合、そのクラスが使用されます。
- その他の条件下では、システムのデフォルトのスケジューリングクラスが使用されます。

dedicated-cpu: ncpus、importance
このゾーンが排他的に使用するために割り当てられるべき CPU の数。ゾーンは、ブート時にプールおよびプロセッサのセットを作成します。リソースプールの詳細については、pooladm(1M) および poolcfg(1M) を参照してください。ncpu プロパティーには、プロセッサ数を単一の値か範囲 (たとえば 1-4) で指定できます。importance プロパティーは省略可能であり、これが設定されると、poold(1M) によって使用される pset.importance の値を指定します。このリソースを使用する場合、ゾーンに割り当てるのに十分な数の空きプロセッサがゾーンのブート時に必要であり、足りないとゾーンはブートしません。この

ゾーンに割り当てられたプロセッサを、大域ゾーンやほかのゾーンが使用することはできません。このリソースには、pool および cpu-shares プロパティーとの互換性はありません。このリソースの単一のインスタンスのみをゾーンに追加できます。

capped-memory: physical、swap、locked
このゾーンが使用できるメモリーの上限。これらの各数値には、K、M、G、T の単位を適用できます。たとえば、1M は 1 メガバイトです。これらの各プロパティーは省略可能ですが、このリソースを追加するときは少なくとも 1 つのプロパティーを設定する必要があります。このリソースの単一のインスタンスのみをゾーンに追加できます。physical プロパティーは、このゾーンの max-rss を設定します。これは大域ゾーンで実行中の rcapd(1M) によって強制されます。swap プロパティーは、zone.max-swap rctl を設定するための望ましい方法です。locked プロパティーは、zone.max-locked-memory rctl を設定するための望ましい方法です。

capped-cpu: ncpus
ゾーンで使用できる CPU 時間に対して制限を設定します。使用される単位は、ゾーン内のすべてのユーザースレッドが使用できる 1 つの CPU の割合 (パーセント) に換算され、小数 (たとえば .75) または帯小数 (整数と小数が混在した数、たとえば 1.25) で表現されます。ncpu 値 1 は CPU の割合が 100% であることを意味し、値 1.25 は 125%、.75 は 75% (以下同様) を意味します。上限が設けられたゾーン内のプロジェクトに独自の上限がある場合は、最小値が優先されます。

capped-cpu プロパティーは、zone.cpu-cap リソース制御の別名であり、zone.cpu-cap リソース制御に関連しています。resource_controls(5) を参照してください。

admin: user、auths
ゾーン管理の承認を、指定されたユーザーまたは役割に委任します。ユーザーは有効なローカルアカウントに対応している必要があります。auths に使用できる値は次のとおりです。

login
このゾーンへの zlogin(1) の使用の認証を許可します。

manage
構成済みゾーンの通常の管理を許可します。

copyfrom
新しいゾーンのクローンを作成するソースとして、指定されたゾーンの使用を許可します。

次の表は、リソース、プロパティー名、および型の一覧です。

resource	property-name	type
(global)	zonename	simple
(global)	zonepath	simple

```
(global)          autoboot        simple
(global)          bootargs        simple
(global)          pool            simple
(global)          limitpriv       simple
(global)          brand           simple
(global)          ip-type         simple
(global)          hostid          simple
(global)          cpu-shares      simple
(global)          max-lwps        simple
(global)          max-msg-ids     simple
(global)          max-sem-ids     simple
(global)          max-shm-ids     simple
(global)          max-shm-memory  simple
(global)          scheduling-class simple
(global)          fs-allowed      list of simple
fs                dir             simple
                  special         simple
                  raw             simple
                  type            simple
                  options         list of simple
net               address         simple
                  allowed-address list of simple
                  configure-allowed-address simple
                  defrouter       list of simple
                  physical        simple
anet              linkname        simple
                  lower-link      simple
                  allowed-address list of simple
                  configure-allowed-address simple
                  defrouter       list of simple
                  mac-address     simple
                  mac-slot        simple
                  mac-prefix      simple
                  mtu             simple
                  maxbw           simple
                  priority        simple
                  vlan-id         simple
                  rxfanout        simple
                  rxrings         simple
                  txrings         simple
                  link-protection list of simple
                  allowed-dhcp-cids list of simple
device            match           simple
rctl              name            simple
                  value           list of complex
attr              name            simple
                  type            simple
                  value           simple
```

```
dataset          name           simple
                 alias          simple
dedicated-cpu    ncpus          simple or range
                 importance     simple

capped-memory    physical       simple with scale
                 swap           simple with scale
                 locked         simple with scale

capped-cpu       ncpus          simple

admin            user           simple
                 auths          list of simple
```

この表について、いくつか説明を補足します。「rctl」リソースタイプの複合プロパティー「value」は、名前/値の 3 つの組み合わせで構成されます。名前は「priv」、「limit」、および「action」で、それぞれ単純値を指定します。「attr」リソースの「name」プロパティーの構文には、ゾーン名と同じような制限がありますが、相違点もあります。 つまり、英数字で始める必要がありますが、英数字以外にハイフン (-)、下線 (_)、およびピリオド (.) も使用できます。「zone」で始まる属性名はシステム用に予約されています。また、「autoboot」大域プロパティーの値は、必ず「true」または「false」にしてください。

カーネル統計を使った CPU キャップの監視

カーネル統計 (kstat(3KSTAT)) モジュール caps を使用して、システムは上限が設定されたすべてのプロジェクトおよびゾーンの情報を維持管理します。この情報を入手するには、kstat モジュール名として caps を指定して、カーネル統計 (kstat(3KSTAT)) を読み取ります。次のコマンドは、アクティブなすべての CPU キャップのカーネル統計を表示します。

```
# kstat caps::'/cpucaps/'
```

ゾーン内で実行されている kstat(1M) コマンドは、そのゾーンとそのゾーン内のプロジェクトに関連する CPU キャップしか表示しません。「使用例」を参照してください。

kstat(1M) で使用するキャップ関連の引数を次に示します。

caps
: kstat モジュール。

project_caps または zone_caps
: kstat クラス。kstat -c オプションとともに使用します。

cpucaps_project_*id* または cpucaps_zone_*id*
: kstat 名。kstat -n オプションとともに使用します。*id* はプロジェクトまたはゾーンの識別子です。

次のフィールドは、すべての CPU キャップの統計情報を要求する kstat(1M) コマンドに応答して表示されます。

module
kstat のこの使用法では、このフィールドには値 caps が入ります。

name
前述のとおり、cpucaps_project_*id* または cpucaps_zone_*id*

above_sec
キャップを上回って費やされた合計時間 (秒)。

below_sec
キャップを下回って費やされた合計時間 (秒)。

maxusage
監視された最大 CPU 使用率。

nwait
待機キューに入っているスレッドの数。

usage
上限が設けられたプロジェクトまたはゾーンに属しているすべてのスレッドの現在の合計 CPU 使用率。1 つの CPU の割合 (パーセント) で表されます。

value
キャップ値。1 つの CPU の割合 (パーセント) で表されます。

zonename
統計情報が表示されるゾーンの名前。

kstat コマンドの出力例については、「使用例」を参照してください。

オプション

次のオプションがサポートされています。

-f *command_file*
zonecfg コマンドファイルの名前を指定します。*command_file* は、zonecfg サブコマンドを 1 行に 1 つずつ記述するテキストファイルです。

-z *zonename*
ゾーンの名前を指定します。ゾーン名では、大文字と小文字が区別されます。ゾーン名は英数字で始める必要がありますが、英数字以外に下線 (_)、ハイフン (-)、およびピリオド (.) も使用できます。global という名前と SYS で始まるすべての名前は、予約されているので使用できません。

サブコマンド

add および select サブコマンドを使用して、特定のリソースを選択できます。選択した時点で、有効範囲がそのリソースに変更されます。end および cancel サブコマンドを使用して、そのリソースの指定を終了することができます。指定を終了した時点で、有効範囲が大域に戻ります。add、remove、set などの一部のサブコマンドでは、有効範囲ごとにセマンティクスが異なります。

zonecfg は、サブコマンドをセミコロンで区切ったリストをサポートします。例:

```
# zonecfg -z myzone "add net; set physical=myvnic; end"
```

破壊的な動作や作業内容の消失を伴う可能性のあるサブコマンドには、強制的にその処理を実行するために -F オプションが用意されています。端末デバイスから入力しているときに、-F オプションを指定しないでそのようなコマンドを実行した場合は、オプションを指定するかどうかを適切なタイミングで確認されます。それ以外の状況で -F オプションを指定しないでそのようなコマンドを実行した場合、その操作は許可されず、診断メッセージが標準エラーに書き込まれます。

サポートされているサブコマンドは次のとおりです。

add *resource-type* (global scope)
add *property-name property-value* (resource scope)
: 大域有効範囲の場合は、特定のリソースタイプの指定を開始します。有効範囲がそのリソースタイプに変わります。

  リソース固有の有効範囲では、指定された名前と値を持つプロパティーを追加します。プロパティー値の構文は、プロパティータイプによって異なります。通常は、単純値または単純値のリストを使用します。リストの場合は、角括弧で囲み、各値をコンマで区切ります ([foo,bar,baz])。「プロパティー」を参照してください。

cancel
: リソースの指定を終了し、有効範囲を大域に戻します。指定途中のリソースをすべて破棄します。cancel は、リソースの有効範囲だけに適用できます。

clear *property-name*
: プロパティーの値を消去します。

commit
: 現在の構成をメモリーから安定した記憶領域に確定します。zoneadm で構成を使用するには、その構成を確定する必要があります。メモリー内構成を確定するまでは、revert サブコマンドを使って変更を取り消すことができます。zonecfg セッションが終了するときには、commit 処理が自動的に実行されます。構成が正しく設定されていないと確定できないので、この処理では自動的に構成が確認されます。

create [-F] [-a *path* |-b | -t *template*]
: 指定されたゾーンのメモリー内構成を作成します。新しいゾーンの構成を開始するときは、create を使用します。このゾーンを安定した記憶領域に保存する方法については、commit を参照してください。

  既存の構成を上書きする場合は、-F オプションを指定して、強制的に処理を実行します。*template* と同じ構成を作成する場合は、-t *template* オプションを指定します。*template* は設定済みゾーンの名前です。

create はデフォルトテンプレート SYSdefault を使用します。デフォルトテンプレートは、svc:/system/zones:default サービスの default_template SMF プロパティーを使用することによって、システム全体ベースで変更できます。管理者は、-t (特定のテンプレートを使用) または -b (ブランクテンプレートを使用) を使用することによって、このゾーンのデフォルトを置き換えることができます。

新しいホストで切り離されたゾーンを構成できるようにするには、-a *path* オプションを使用します。*path* パラメータは、この新しいホスト上に移動されている、切り離されたゾーンのゾーンパスの位置です。切り離されたゾーンが構成されたあとは、「zoneadm attach」コマンド (zoneadm(1M) を参照) を使用してインストールするようにしてください。新しいゾーンのすべての検証は、ゾーンの構成中ではなく attach 処理中に行われます。

空の構成を作成する場合は、-b オプションを使用します。引数が指定されていない場合、create は Oracle Sun のデフォルト設定を適用します。

delete [-F]
: 指定された構成をメモリー内および安定した記憶領域から削除します。この操作はすぐに実行されるので、確定する必要はありません。削除した構成を元に戻すことはできません。

この操作を強制的に実行する場合は、-F オプションを使用します。

end
: リソースの指定を終了します。このサブコマンドは、リソースの有効範囲だけに適用できます。zonecfg は、現在のリソースが完全に指定されていることを確認します。完全に指定されている場合は、メモリー内構成に追加され (メモリー内構成を安定した記憶領域に保存する方法については commit を参照)、有効範囲は大域に戻ります。指定が完全でない場合は、対応するエラーメッセージが通知されます。

export [-f *output-file*]
: 標準出力に構成を出力します。*output-file* に構成を出力する場合は、-f オプションを使用します。このオプションを使用すると、コマンドファイルに適した形式で出力が作成されます。

help [usage] [*subcommand*] [syntax] [*command-name*]
: 一般ヘルプまたは特定項目のヘルプを出力します。

info zonename | zonepath | autoboot | brand | pool | limitpriv
info [*resource-type* [*property-name*=*property-value*]*]
: 現在の構成に関する情報を表示します。*resource-type* が指定されている場合は、そのタイプのリソースについてのみ情報を表示します。*property-name* 値ペアが指定されている場合は、その条件を満たしているリソースの情報だけを表示します。リソースの有効範囲では、引数が無視され、info は現在追加または変更されているリソースの情報を表示します。

`remove` *resource-type*{*property-name*=*property-value*}(global scope)
大域有効範囲では、指定されたリソースタイプを削除します。[] 構文は、0 個以上の項目を角括弧の中に指定できることを意味します。リソースの単一インスタンスを削除するだけの場合、そのリソースが一意に識別されるように、プロパティーの名前と値ペアを十分に指定する必要があります。プロパティーの名前と値の対をまったく指定しないと、すべてのインスタンスが削除されます。2 つ以上のペアを指定した場合は、`-F` オプションを指定しないかぎり、確認を要求されます。

`select` *resource-type* {*property-name*=*property-value*}
指定されたリソースタイプのうち、指定された *property-name* と *property-value* の対の条件に一致するものを、変更対象として選択します。このサブコマンドは、大域有効範囲だけに適用できます。有効範囲がそのリソースタイプに変わります。{} 構文は、1 つまたは複数の項目をこの中に指定できることを意味します。リソースが一意に識別されるように、*property-name property-value* ペアを十分に指定する必要があります。

`set` *property-name*=*property-value*
指定されたプロパティー名を、指定された値に設定します。プロパティーには、`zonename` や `zonepath` のような大域プロパティーと、リソース固有のプロパティーがあります。このサブコマンドは、大域有効範囲とリソース固有の有効範囲の両方で使用できます。

`verify` [`-v`]
現在の構成が正しいかどうかを確認します。

- 各リソースに必須プロパティーがすべて指定されていること。
- `zonepath` が指定されていること。

`-v` オプションが指定された場合、デバイスリソース内で指定されたデバイスが、別名が付けられたデータセット内で作成された ZFS ボリュームと競合して、ZFS ボリュームを非表示にする可能性があるときに、警告が発行されます。dev(7FS) を参照してください。

`revert` [`-F`]
構成を、最後に確定されたときの状態に戻します。この操作を強制的に実行する場合は、`-F` オプションを使用します。

`exit` [`-F`]
`zonecfg` のセッションを終了します。必要な場合は、確定操作が自動的に試行されます。EOF 文字を使用して `zonecfg` を終了することもできます。この操作を強制的に実行する場合は、`-F` オプションを使用します。

使用例

例 1　新しいゾーンの環境を作成する

次の例では、`zonecfg` を使って新しいゾーンの環境を作成します。`/usr/local` は、大域ゾーンから `/opt/local` にマウントされるループバックです。`/opt/sfw` は、大域ゾーンからマウントされるループバックで、`nxge0` 上の VNIC が 3 つの IP アドレスとともにこのゾーンに追加され、ゾーンに対する公平配分スケジューラ

例 1　新しいゾーンの環境を作成する　　*(続き)*

(FSS) の CPU 共有の数に関する制限が、rctl リソースタイプを使用して設定されます。この例では、所定のリソースを変更するために選択する方法も示されています。このケースでは、zonecfg によって自動的に作成される anet リソースを選択することによって実行します。

```
example# zonecfg -z myzone3
my-zone3: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:myzone3> create
zonecfg:myzone3> set zonepath=/export/home/my-zone3
zonecfg:myzone3> set autoboot=true
zonecfg:myzone3> add fs
zonecfg:myzone3:fs> set dir=/usr/local
zonecfg:myzone3:fs> set special=/opt/local
zonecfg:myzone3:fs> set type=lofs
zonecfg:myzone3:fs> add options [ro,nodevices]
zonecfg:myzone3:fs> end
zonecfg:myzone3> add fs
zonecfg:myzone3:fs> set dir=/mnt
zonecfg:myzone3:fs> set special=/dev/dsk/c0t0d0s7
zonecfg:myzone3:fs> set raw=/dev/rdsk/c0t0d0s7
zonecfg:myzone3:fs> set type=ufs
zonecfg:myzone3:fs> end
zonecfg:myzone3> add fs
zonecfg:myzone3:fs> set dir=/opt/sfw
zonecfg:myzone3:fs> set special=/opt/sfw
zonecfg:myzone3:fs> set type=lofs
zonecfg:myzone3:fs> add options [ro,nodevices]
zonecfg:myzone3:fs> end
zonecfg:myzone3> select anet linkname=net0
zonecfg:myzone3:anet> set lower-link=nxge0
zonecfg:myzone3:anet> set allowed-address="192.168.0.1/24,192.168.1.2/\
           24,192.168.2.3/24"
zonecfg:myzone3:anet> end
zonecfg:my-zone3> set cpu-shares=5
zonecfg:my-zone3> add capped-memory
zonecfg:my-zone3:capped-memory> set physical=50m
zonecfg:my-zone3:capped-memory> set swap=100m
zonecfg:my-zone3:capped-memory> end
zonecfg:myzone3> exit
```

例 2　排他的 IP ゾーンを作成する

次の例では、net0 という名前の VNIC が割り当てられるゾーンを作成します。VNIC が作成されるリンクは自動的に決定されます。IP アドレスおよびルーティングは、新しいゾーンの中で ipadm(1M) を使用して構成されます。

例 2　排他的 IP ゾーンを作成する　　*(続き)*

```
example# zonecfg -z excl
zonecfg:excl> create
zonecfg:excl> set zonepath=/export/zones/excl
zonecfg:excl> exit
```

例 3　共有 IP ゾーンを作成する

次の例では、IP スタックを大域ゾーンと共有し、単一の IP アドレスとデフォルトルーターが割り当てられるゾーンを作成します。

```
example# zonecfg -b -z shared
zonecfg:shared> create
zonecfg:shared> set zonepath=/export/zones/shared
zonecfg:shared> set ip-type=shared
zonecfg:shared> add net
zonecfg:shared:net> set physical=nge0
zonecfg:shared:net> set address=192.168.0.3/24
zonecfg:shared:net> set defrouter=192.168.0.1
zonecfg:shared:net> end
zonecfg:shared> exit
```

例 4　ゾーンをリソースプールに関連付ける

次の例は、既存のゾーンを既存のリソースプールに関連付ける方法を示しています。

```
example# zonecfg -z myzone
zonecfg:myzone> set pool=mypool
zonecfg:myzone> exit
```

リソースプールの詳細については、pooladm(1M) および poolcfg(1M) を参照してください。

例 5　ゾーンの名前を変更する

次の例は、既存のゾーンの名前を変更する方法を示したものです。

```
example# zonecfg -z myzone
zonecfg:myzone> set zonename=myzone2
zonecfg:myzone2> exit
```

例 6　ゾーンの特権セットを変更する

次の例では、次回ゾーンのブート時に制限される既存のゾーンのプロセスの特権セットを変更する方法を示しています。この場合の特権セットは、システムの日付と時間を変更する特権に加え、通常ゾーンが持っている標準的な安全特権セットになります。

例 6 ゾーンの特権セットを変更する (続き)

```
example# zonecfg -z myzone
zonecfg:myzone> set limitpriv="default,sys_time"
zonecfg:myzone2> exit
```

例 7 大域ゾーンの zone.cpu-shares プロパティーを設定する

次のコマンドでは、大域ゾーンの zone.cpu-shares プロパティーを設定します。

```
example# zonecfg -z global
zonecfg:global> set cpu-shares=5
zonecfg:global> exit
```

例 8 パターンマッチングを使用する

次のコマンドでは、zonecfg のパターンマッチングのサポートを説明しています。ゾーン flexlm で、次のように入力します。

```
zonecfg:flexlm> add device
zonecfg:flexlm:device> set match="/dev/cua/a00[2-5]"
zonecfg:flexlm:device> end
```

大域ゾーンで、次のように入力します。

```
global# ls /dev/cua
a     a000  a001  a002  a003  a004  a005  a006  a007  b
```

ゾーン flexlm で、次のように入力します。

```
flexlm# ls /dev/cua
a002  a003  a004  a005
```

例 9 ゾーンのキャップを 3 つの CPU に設定する

次のシーケンスでは、zonecfg コマンドを使ってゾーンの CPU キャップを 3 つの CPU に設定します。

```
zonecfg:myzone> add capped-cpu
zonecfg:myzone>capped-cpu> set ncpus=3
zonecfg:myzone>capped-cpu>capped-cpu> end
```

前のシーケンスでは capped-cpu プロパティーを使用し、次のシーケンスでは zone.cpu-cap リソース制御を使用していますが、これらのシーケンスは同等です。

```
zonecfg:myzone> add rctl
zonecfg:myzone:rctl> set name=zone.cpu-cap
zonecfg:myzone:rctl> add value (priv=privileged,limit=300,action=none)
zonecfg:myzone:rctl> end
```

例 10 kstat を使って CPU キャップを監視する

次のコマンドは、すべての CPU キャップに関する情報を表示します。

例 10　kstat を使って CPU キャップを監視する　　(続き)

```
# kstat -n /cpucaps/
module: caps                            instance: 0
name:   cpucaps_project_0               class:    project_caps
        above_sec                       0
        below_sec                       2157
        crtime                          821.048183159
        maxusage                        2
        nwait                           0
        snaptime                        235885.637253027
        usage                           0
        value                           18446743151372347932
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_1               class:    project_caps
        above_sec                       0
        below_sec                       0
        crtime                          225339.192787265
        maxusage                        5
        nwait                           0
        snaptime                        235885.637591677
        usage                           5
        value                           18446743151372347932
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_201             class:    project_caps
        above_sec                       0
        below_sec                       235105
        crtime                          780.37961782
        maxusage                        100
        nwait                           0
        snaptime                        235885.637789687
        usage                           43
        value                           100
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_202             class:    project_caps
        above_sec                       0
        below_sec                       235094
        crtime                          791.72983782
        maxusage                        100
        nwait                           0
        snaptime                        235885.637967512
        usage                           48
```

例 10　kstat を使って CPU キャップを監視する　　(続き)

```
        value                           100
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_203             class:    project_caps
        above_sec                       0
        below_sec                       235034
        crtime                          852.104401481
        maxusage                        75
        nwait                           0
        snaptime                        235885.638144304
        usage                           47
        value                           100
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_86710           class:    project_caps
        above_sec                       22
        below_sec                       235166
        crtime                          698.441717859
        maxusage                        101
        nwait                           0
        snaptime                        235885.638319871
        usage                           54
        value                           100
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_zone_0                  class:    zone_caps
        above_sec                       100733
        below_sec                       134332
        crtime                          821.048177123
        maxusage                        207
        nwait                           2
        snaptime                        235885.638497731
        usage                           199
        value                           200
        zonename                        global

module: caps                            instance: 1
name:   cpucaps_project_0               class:    project_caps
        above_sec                       0
        below_sec                       0
        crtime                          225360.256448422
        maxusage                        7
        nwait                           0
```

例 10　kstat を使って CPU キャップを監視する　　(続き)

```
        snaptime                        235885.638714404
        usage                           7
        value                           18446743151372347932
        zonename                        test_001

module: caps                            instance: 1
name:   cpucaps_zone_1                  class:    zone_caps
        above_sec                       2
        below_sec                       10524
        crtime                          225360.256440278
        maxusage                        106
        nwait                           0
        snaptime                        235885.638896443
        usage                           7
        value                           100
        zonename                        test_001
```

例 11　特定のゾーンまたはプロジェクトの CPU キャップを表示する

kstat -c および -i オプションを使用すると、次に示すように特定のゾーンまたはプロジェクトの CPU キャップを表示できます。最初のコマンドは特定のプロジェクトを対象とした表示を行い、2 番目のコマンドはゾーン 1 内部の同じプロジェクトを対象とした表示を行います。

```
# kstat -c project_caps

# kstat -c project_caps -i 1
```

例 12　ゾーン管理権を委任する

次の例は、現在のゾーンの管理権を役割に割り当てる方法を示しています。

```
example# zonecfg -z myzone
zonecfg:myzone> add admin
zonecfg:myzone:admin> set user=zadmin
zonecfg:myzone:admin> set auths=login,manage
zonecfg:myzone:admin> end
zonecfg:myzone> commit
```

これらのコマンドを実行すると、RBAC user_attr(4) データベース内で次のようにエントリが更新されます。

```
zadmin::::type=role;\
auths=solaris.zone.login/myzone,solaris.zone.manage/myzone;profiles=\
Zone Management
```

例 13 デフォルト以外のプロパティーを使用して排他的 IP ゾーンを作成する

次の例はゾーンを作成し、mylink0 上に指定の MAC アドレスを持ち、最大帯域幅が 100Mbps、高い優先順位、RX 側に専用ハードウェアリング、TX 側に専用ハードウェアリングなし (つまりソフトウェアベース)、および VLAN ID が 2 の VNIC が自動的に作成されます。

```
example# zonecfg -b -z excl
excl: No such zone configured
Use 'create' to begin configuring a new zone
zonecfg:excl> create
zonecfg:excl> set zonepath=/export/zones/excl
zonecfg:excl> add anet
zonecfg:excl:anet> set linkname=mynic0
zonecfg:excl:anet> set lower-link=mylink0
zonecfg:excl:anet> set mac-address=8:0:20:fe:4e:b8
zonecfg:excl:anet> set maxbw=100M
zonecfg:excl:anet> set priority=high
zonecfg:excl:anet> set vlan-id=2
zonecfg:excl:anet> set rxrings=hw
zonecfg:excl:anet> set txrings=sw
zonecfg:excl:anet> end
zonecfg:excl> exit
```

例 14 読み取り専用ゾーンを作成する

次の例では、ルートファイルシステムがゾーンによる変更から保護される新しいゾーンを作成します。/var 内のファイルは、fixed-configuration プロファイルが適用されるため、書き込み可能です。

```
example# zonecfg -z rozone
rozone: No such zone configured
Use 'create' to begin configuring a new zone
zonecfg:rozone> create
zonecfg:rozone> set brand=solaris
zonecfg:rozone> set zonepath=/export/zones/rozone
zonecfg:rozone> set autoboot=true
zonecfg:rozone> set file-mac-profile=fixed-configuration
zonecfg:rozone> set ip-type=exclusive
zonecfg:rozone> add net
zonecfg:rozone:net> set physical=vnic0
zonecfg:rozone:net> end
zonecfg:rozone> exit
```

終了ステータス

次の終了値が返されます。

0
正常終了。

1
エラーが発生しました。

2
無効な使用法。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	流動的

関連項目

ppriv(1), prctl(1), zlogin(1), dladm(1M), ipadm(1M), kstat(1M), mount(1M), pooladm(1M), poolcfg(1M), poold(1M), rcapd(1M), rctladm(1M), route(1M), svcadm(1M), zfs(1M), zoneadm(1M), zpool(1M), priv_str_to_set(3C), kstat(3KSTAT), user_attr(4), vfstab(4), attributes(5), brands(5), fnmatch(5), mwac(5), privileges(5), rbac(5), resource_controls(5), zones(5), dev(7FS), hsfs(7FS), uscsi(7I)

『Oracle Solaris のシステム管理 (Oracle Solaris ゾーン、Oracle Solaris 10 ゾーン、およびリソース管理)』

注意事項

zonecfg で使用するすべての文字データは、US-ASCII エンコーディングにする必要があります。

一般に、すべてのデバイスに対してゾーン内で安全に使用できるどうかの監査が行われるわけではないため、ゾーンにデバイスを追加すると、そのゾーンがシステムのセキュリティーおよび安定性に悪影響を与えることがあります。

sd または ssd ターゲットドライバを使用するストレージデバイス (prtconf -D /dev/dsk/c2t40d3 などを使用すると検査できます) は、ゾーンに安全に委任できます。このため、ゾーン管理者は、このようなデバイスに対してラベル付けやパーティション分割を実行できます。

スライス (たとえば、match=/dev/dsk/c2t40d3s0) だけを委任した場合も、ゾーン管理者は、そのスライスを含むディスク全体に対してラベル付けやパーティション分割を実行できるようになります。これを回避するには、format を実行できる機能を次のゾーンから削除してください。

```
set limitpriv=default,-storage_config
```

raw_storage 特権をゾーンの limitpriv に追加すると、ストレージデバイスへの raw アクセスを有効にできます。ゾーンプロセスで raw SCSI コマンド (uscsi(7I) を参照) を実行できるようになるため、この操作は安全ではありません。

名前

zonep2vchk – check a global zone's configuration for physical to virtual migration into non-global zone

形式

```
zonep2vchk -V

zonep2vchk [-T release] -c

zonep2vchk [-T release] [-P] [-b] [ -s path[,path...] ] [-S file]
     [ -r {time}(h|m|s} ] [-x] [-e execname[,execname...] ]
     [-E file]
```

機能説明

The `zonep2vchk` utility is used to evaluate a global zone's configuration before the process of physical-to-virtual (`p2v`) migration into a non-global zone.

The p2v process involves archiving a global zone (source), and then installing a non-global zone (target) using that archive. See the `install -a` documentation in the `solaris(5)` and `solaris10(5)` man pages.

`zonep2vchk` serves two functions. First, it can be used to report issues on the source which might prevent a successful `p2v` migration. Second, it can output a template `zonecfg`, which can be used to assist in configuring the non-global zone target.

`zonep2vchk` can be executed on a Solaris 10 or later global zone. To execute on Solaris 10, copy the `zonep2vchk` utility to the Solaris 10 source global zone.

セキュリティー

The `zonep2vchk` utility must be run with an effective user id of zero. It interrogates the configuration state of a variety of Solaris subsystems.

オプション

The following options are supported:

`-V`
: Display the command version and exit.

`-T` *release*
: Specify the target release. The defaults are:

```
Global Zone         Default Target
Solaris 10          S10
Solaris 11          S11
```

Any configuration files generated by `zonep2chk` will be applicable to the target release. See `-c` below.

When run on Solaris 10, a target release of `S11` can be specified, which will check for `p2v` into a Solaris 10 Branded zone.

When the target is S10, it is assumed that a shared stack will be used. Any issues that will require an exclusive IP stack will be reported.

When the target is S11, it is assumed that an exclusive IP stack will be used.

If a particular feature in use by the global zone requires a particular patch/update level of the target to function, this information will be printed in the `zonep2vchk` output.

`-P`
Generate machine-parseable output. See the section "Parseable Output Format" below.

`-c`
Display a template zone configuration on stdout in the form of `zonecfg(1M)` `export` output. This configuration will contain resource limits and network configuration based on the source host's physical resources and networking configuration.

`-b`
Perform basic checks. This will check the global zone for issues that could prevent a successful `p2v`. This is the default behavior if none of `-b`, `-c`, `-s`, `-S`, `-r`, `-x` are specified.

`-r`{*time*} (h|m|s)
Perform runtime checks for the specified duration. This will analyze the currently executing processes in the global zone, and report issues that could prevent successful execution inside a non-global zone. Issues reported reflect actions made by the processes during the time in which `zonep2vchk` was executing.

`-x`
Perform runtime checks (as with `-r`) until `SIGINT` is received, such as is delivered by `Ctrl-c` from most shells.

`-e` *execname*[,*execname*...]
When performing runtime analysis (`-r`, `-x`), limit inspected programs to those matching the specified list of *execname*s. The *execname* is the name of process, as returned by `ps -o comm`. It is not necessary for named processes to exist when `zonep2vchk` is invoked. Any matching processes created while `zonep2vchk` is running will be inspected.

`-E` *file*
Similar to `-e`, but reads the list of *execname*s from *file*, one per line.

`-s` *path*[,*path*...]
Perform static binary analysis on the files or directories specified. This will inspect ELF binaries for system and library calls that might affect function inside a zone. Directories will be recursed, and non-ELF files will be ignored.

`-S` *file*
Similar to `-s`, but reads the path list from *file*, one per line.

Parseable Output Format

`zonep2vchk` will ouput a single line of parseable output for each issue detected. The line format is:

category:*issue*:*field1*:[*field2*:...]

Each field is delimited by a colon (`:`). Colon characters escaped with a backslash (`\:`) should not be treated as field delimiters.

Multiple instances of the same issue can be reported, each with fields describing the particular instance of the issue.

Below the existing categories and issues are defined. Future versions of `zonep2vchk` might include additional categories and issues. Existing issues might have new fields added after the existing fields for existing issues.

header Category

The `header` category lists information about the source, target, and `zonep2vchk` version. The issues in this category are:

`version`
: The version of the `zonep2vchk` command.

Field1: The version of the `zonep2vchk` command.

`source`
: Information about the source system.

Field1: The nodename of the source system.

Field2: The /etc/release version of the source system.

Field3: The kernel version of the source system.

Field4: The platform of the source system.

`target`
: Information about the specified target of the `p2v` check.

Field1: The Solaris version of the target.

Field2: The brand type that would be used on the target.

Field3: The ip-type of the expected zone on the target.

footer Category

The `footer` category lists final summary information. The issues in this category are:

`issues`
: A summary of the number of issues found.

Field1: The number of issues detected.

incompatible Category

The `incompatible` category represents issues that will not function in a non-global zone. The issues in this category are:

`etcsystem`
: An `/etc/system` tunable exists. These tunables do not function inside a zone. The `/etc/system` tunable can be transferred to the target global zone, but it will affect the entire system, including all zones and the global zone. If there is an alternate tunable that can be configured from within the zone, this tunable is described.

Field1:
: The `/etc/system` tunable setting.

Field2:
One of:

`noalternate` There is no alternate tunable from within a non-global zone.

`obsolete` The tunable is obsolete on the target. It no longer serves any function.

`replaced` The tunable has been replaced on the target. The replacement is configured in the global zone, and described by fields 3 and 4.

`alternate` An alternate tunable exists. This tunable can be configured from within a non-global zone. The tunable is described by fields 3 and 4.

`noinfo` `zonep2vchk` is not knowledgeable of the tunable. Tunable likely has no alternate inside a zone.

Field3:
Type of alternate/replacement tunable.

Field4:
Description of alternate/replacement tunable.

`be`
More than one boot environment exists. Only the active boot environment will be transferable to the non-global zone.

Field1: The name of the non-active boot environment.

`unsupported`
A feature is enabled that will not function in a zone.

Field1: `mobileip` The mobile IP agent, which does not function in a zone, is configured. See `mipagent(1M)` for details.

`nfs` (S10 sources only)
The system is sharing a filesystem by means of NFS. Native zones on Solaris 10 and Solaris 10 zones on Solaris 11 cannot share by means of NFS.

Field1: Path of file system being shared.

`smb`
The system is sharing a filesystem by means of in-kernel `smb/cifs`. Zones cannot share filesystems by means of SMB.

Field1: Path of file system being shared.

`pkg`
A package delivering software known not to work in a zone is installed.

Field1: Name of the package.

`iscsi-target`
The system is exporting an ISCSI target. Zones cannot export iSCSI targets.

Field1: Name of the iSCSI target.

`fcoe-target`
The system has configured an FCOE target. Zones cannot configure FCOE targets.

Field1: Ethernet device used.

Field2: WWN of the FCOE target.

`fc-target`
The system has configured an Fiberchannel target. Zones cannot configure Fiberchannel targets.

Field1: WWN of the Fiberchannel target.

`npiv`
The system has configured a virtual NPIV HBA. Zones cannot configure virtual HBAs.

Field1: Physical WWN hosting the virtual HBA.

Field2: Virtual WWN.

`scsi`
The system has configured an SCSI block device. Zones cannot configure scsi block devices.

Field1: Object configured as a SCSI device.

`svcnotallowed`
A service is enabled that will not function in a zone.

Field1: Name of the service.

`resourcepool`
A Solaris resource pool is configured. Zones cannot configure resource pools.

Field1: Name of the pool.

`pset`
A processor set is configured. Zones cannot configure processor sets.

Field1: Processor set ID.

Field2: List of CPU IDs in the processor set.

`zones`
Zones are configured. A zone cannot host zones. Any zones will not exist in the target non-global zone after `p2v`. Zones can be migrated separately using the `detach`/`attach` features in `zoneadm`(1M).

Field1: Name of the zone.

Field2: State of the zone.

`lofi` (Solaris 10 targets only)
A lofi device is configured. A zone cannot configure lofi devices.

Field1: Name of the lofi device.

Field2: Path of the file backing the device.

`syscall` (generated by `-s` and `-f`)
A binary makes a system or library call that cannot be made from a zone.

Field1: Name of the the binary file.

Field2: Name of the system or library call.

`syscallargs` (generated by `-s` and `-f`)
A binary makes a system or library call that cannot be made from a zone if called with certain arguments.

Field1: Name of the system or library call.

See regular output (no `-P`) for details on disallowed arguments.

`lib` (generated by `-s` and `-f`)
A binary links with a library that cannot be used inside a zone.

Field1: Name of the binary file.

Field2: Name of the disallowed library.

`privnotallowed` (generated by `-r` and `-x`)
A privilege is used by a process that cannot be added to a zone.

Field1: Name of the process.

Field2: Name of the privilege.

`devnotallowed` (generated by `-r` and `-x`)
A device is opened by a process that cannot be added to a zone.

Field1: Name of the process.

Field2: Name of the device.

configuration Category

The `configuration` category represents issues that will require a configuration setting to allow the issue to function inside the non-global zone. This could be a `zonecfg(1M)` configuration setting, a configuration change in the global zone, or both.

The issues in this category are:

`datalink`
A datalink feature is configured that cannot be configured from within a zone. The datalink feature must be configured in the global zone, and if necessary, delegated to the zone using `zonecfg add anet` (Solaris 11 only) or `zonecfg add net`.

Field1: Name of the datalink feature. One of:

`aggr`	Aggregation.
`ibiface`	Infiniband interface.
`ibpart`	Infiniband partition.
`vnic`	Virtual NIC.
`etherstub`	Ethernet stub.
`bridge`	A bridge instance.
`secobj`	A wireless WPA or WEB security object.

Field2: Datalink object name.

`dhcp-server` (Solaris 10 targets only)
The host is a DHCP server. To provide DHCP service, a zone must have `ip-type=exclusive`, or have the the privilege `net_rawaccess` and the device `/dev/ip`. Note that this will allow a shared stack zone to read and write raw IP packets on the network, similar to an exclusive stack zone or global zone.

Field1: FMRI of the DHCP server service.

`ntp-client`
An NTP client service is enabled. This service updates the system clock. Since all zones share the same system clock, this service is disabled automatically during `p2v`. If it is desired that the zone update the system clock on the target host, the zone will need the privilege `sys_time`, and the service will need to be enabled inside the zone after `p2v`.

Field1: FMRI of the client service.

`driverconf`
A networking device contains configuration settings in its `.conf` file. Zones cannot configure drivers. The driver must be configured in the global zone. Some network driver settings might be configurable using `dladm(1M)` instead of editing a driver configuration file.

Field1: Path of the configuration file.

`ifname` (Solaris 10 targets only)
An existing configuration file will be impacted by the change of a network device name. For example, an `/etc/hostname.bge0` file will be impacted if the network device given to the target non-global zone is not `bge0`.

Field1: Path of the impacted file.

`iscsi-initiator`
The system is accessing an iSCSI target as a client. Zones cannot access iSCSI targets. The global zone must be the iSCSI initiator. The device can then be added to the zone using `zonecfg add device`.

Field1: iSCSI target being accessed.

`fcoe-initiator`

The system has an FCOE initiator configured. A zone cannot configure an FCOE initiator. The global zone must configure the FCOE initiator, and make the SCSI target devices available to the zone using `zonecfg add fs` or `zonecfg add device`.

Field1: Ethernet network device.

Field2: WWN of the initiator.

`fc-initiator`

The system has an HBA Fiberchannel port online. A zone cannot access a Fiberchannel target. The target must be accessed from the global zone and made available to the zone.

Field1: Fiberchannel HBA port WWN.

`linkprop`

Datalink properties are configured. A zone cannot configure datalink properties. They must be configured from the global zone.

Field1: Name of the datalink.

Field2: Property name

Field3: Property value.

`ndd`

Tunables that cannot be configured by a zone have been configured using `ndd`. These tunables must be configured from the global zone.

Field1: File or script setting the tunable.

Field2: Driver being tuned.

Field3: Tunable parameter.

`dynaddr`

One or more dynamically assigned IP addresses are configured on a network interface. These addresses are not supported with shared-IP zones. These IP addresses could change as a result of MAC address changes. You may need to modify this system's address information on the DHCP server and on the DNS, LDAP, or NIS name servers.

Field1 can be one of:

`dhcp`	Configured DHCP address. In this case, *Field2* is the name of the interface configured for DHCP.
`v6autoconf`	IPv6 stateless address configuration is enabled. In this case, *Field2* is the name of the interface with IPv6 auto configuration.

`rarp` (Solaris 10 source only) Reverse ARP assigned address is enabled. In this case, *Field2* is the name of the interface with reverse ARP enabled.

`patch` (Solaris 10 source with Solaris 11+ target only)
A patch is required before `p2v` into a non-global zone.

Field1: The patch required.

`physif` (Solaris 10 targets only)
A physical interface exists on the source system that will have to be replaced with a dedicated physical or VLAN interface on the destination system if migrating to an exclusive-IP zone.

Field1: Name of the interface on the source system.

`sched`
The system is configured with a default scheduling class. The default scheduling class of a non-global zone can be configured using the `zonecfg set scheduler` property. This will be provided in the `-c` output.

Field1: The configured default scheduling class.

`sharedip` (Solaris 10 targets only)
If migrating to a shared-IP zone, the following networking features will need to be configured from the global zone on behalf of the zone.

Field1 can be one of:

`ipmpgroup` An IPMP group is configured. If IPMP is required, it must be configured from the global zone. In this case, *Field2* is the IPMP group name.

`vni` A virtual network interface is configured. These must be configured from the global zone. In this case, *Field2* is the VNI interface name.

`v4forwarding`
`v6forwarding` IP forwarding (v4 or v6) is configured on an interface. In this case, *Field2* is the interface with IP forwarding configured.

`staticroute` Static routes are configured. Static routes must be configured from the global zone.

`exclusiveonly` (Solaris 10 targets only)
A networking feature is configured that is not supported for use with shared-IP zones. The feature will work without modification in exclusive-IP zones.

Field1:

`iptun` A IPv4, IPv6, or 6to4 tunnel interface has been plumbed.

Field2: Name of the tunnel interface.

`sharedonly`
A networking feature is configured that is not supported in an exclusive-IP zone. When migrating to a shared-IP zone, the feature must be configured in the global zone to support communication.

Field1:

`cgtp` A Carrier Grade Transport Protocol interface has been plumbed.

Field2: Name of the CGTP interface.

`netdevalloc`
A networking feature requires its underlying device be allocated to the zone with the zonecfg(1M) add device command. This feature is not supported with shared-IP zones.

Field1: Can be :

`ppp` Point-to-Point Protocol (PPP). PPP configuration files exist under `/etc/ppp`. The underlying device that needs to be allocated to the zone is either a serial port or, in the case of `pppoe`, an Ethernet physical or VNIC interface.

`svcexlip` (Solaris 10 targets only)
A service is enabled that will require an exclusive-IP zone.

Field1: Name of the service FMRI.

`svcpriv`
A service is enabled that will require additional privileges be added to the zone using the `zonecfg(1M)` `limitpriv` property.

Field1: FMRI of the service.

Field2: List of the privileges required by the service.

`svm`
A Solaris Volume Manager metadevice is configured. Metadevices must be configured in the global zone, and made available to the non-global zone using `zonecfg(1M)` `add device`, `add fs`, or `add dataset`.

Field1: Name of the metadevice.

`ramdisk`
A ramdisk device is configured. A zone cannot configure ramdisk devices.

Field1: Ramdisk device path.

`vfstab`
A filesystem mount is configured by means of `/etc/vfstab`. The filesystem must be migrated to the target global zone and made available to the non-global zone.

Field1: Device being mounted.

Field2: Mountpoint.

`zpool`
The system has additional zpools configured. These zpools must be migrated to the target global zone, and made available to the zone using `zonecfg add dataset` or `zonecfg add fs`.

Field1: Name of the pool.

`privexclip` (Solaris 10 targets only)
A process used a privilege that requires and exclusive-IP stack. See `zonecfg(1M)` for a description of the `ip-type` property.

Field1: Name of the process.

Field2: Privilege used.

`devexclip` (Solaris 10 targets only)
A process opened a device that requires an exclusive IP stack. See `zonecfg(1M)` for a description of the `ip-type` property.

Field1: Name of the process.

Field2: Name of the device.

`privoptional`
A process used a privilege that requires additional privilege be added to the target non-global zone. See `zonecfg(1M)` for a description of the `limitpriv` property.

Field1: Name of the process.

Field2: Privilege used.

`devoptional`
A process opened a device that is not available in a zone by default. See `zonecfg(1M)` for a description of the `add device` resource.

Field1: Name of the process.

Field2: Path of the device.

`syscallpriv` (generated by `-s` and `-f`)
A binary makes a system or library call that might require additional privilege be added to the target non-global zone. See `zonecfg(1M)` for a description of the `limitpriv` property. See the non-parseable output for details concerning the system or library call.

Field1: Path of the binary

Field2: Name of the system call.

`syscallexclip` (generated by `-s` and `-f`)
A binary makes a system or library call that might require an exclusive-ip stack. See `zonecfg(1M)` for a description of the `ip-type` property. See the non-parseable output for details concerning the system or library call.

Field1: Path of the binary

Field2: Name of the system call.

使用例

例 1 Performing Static Binary Analysis

The following command performs static analysis on all ELF binaries in two application directory trees:

```
# zonep2vchk -s /opt/myapplication,/usr/local
```

例 2 Generating a Template for the Target Zone

The following command will generate a template zone configuration for Solaris 11 when run on a Solaris 10 global zone.

```
# zonep2vchk T S11 -c
```

例 3 Analyzing Running Applications for a Period

The following command will analyze the process named `myapplication` for one hour and report any activity that might not function in a zone.

```
# zonep2vchk -s 1h -e myapplication
```

例 4 Performing Basic Checks

The following command will analyze the global zone for configuration and Solaris features in use that might not function in a zone. Each discovered issue will be reported as a single line of parseable output.

```
# zonep2vchk  -bP
```

終了ステータス

The following exit values are returned:

`0`
Successful completion, no issues detected.

`1`
An internal error occurred.

`2`
Invalid usage.

`3`
One or more issues were detected.

属性

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/zones
Interface Stability	See below

Command invocation and parseable output is Committed. Human readable output (default output) is Uncommitted.

関連項目

dladm(1M), mipagent(1M), zoneadm(1M), zonecfg(1M), attributes(5), solaris(5), solaris10(5), zones(5)

注意事項

The static (-s and -f) checks make use of the elfdump(1) utility, which is delivered by the following package:

Solaris 11 developer/base-developer-utilities

Solaris 10 SUNWbtool

The runtime (-r) checks make use of the dtrace(1M) utility, which is delivered by the following package:

Solaris 11 system/dtrace

Solaris 10 SUNWdtrc

名前

zonestatd – ゾーン監視デーモン

形式

/usr/lib/zones/zonestatd

機能説明

zonestatd は、システムのブート中に起動されるシステムデーモンです。このデーモンは、ゾーンによるシステムリソースの使用を監視するほか、psrset psets、pool psets、リソース制御設定などのゾーンおよびシステム構成情報を監視します。

このデーモンは、ゾーン管理ソフトウェアにより自動的に起動されるので、直接起動しないようにしてください。これは、プログラミングインタフェースの構成要素ではありませんが、非公開インタフェースに分類されます。

セキュリティー

各非大域ゾーン (NGZ) 内の zonestat サービスを適切に機能させるには、大域ゾーン内の zonestat サービスをオンラインにする必要があります。各 NGZ の zonestat サービスは、システム構成および使用状況データを直接読み取るのではなく、大域ゾーン上の zonestat サービスから読み取ります。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	非公開

関連項目

prctl(1), svcs(1), zonestat(1), acctadm(1M), pooladm(1M), poolcfg(1M), rcapadm(1M), svcadm(1M), attributes(5), smf(5), zones(5)

注意事項

zonestat サービスは、サービス管理機能 smf(5) によって、次のサービス識別子として管理されます。

```
svc:/system/zones-monitoring:default
```

有効化、無効化、または再起動要求など、このサービスに関する管理操作は、svcadm(1M) を使用して実行できます。サービスステータスを照会するには、svcs(1) コマンドを使用します。

zonestat サービスには、次の SMF 構成プロパティーがあります。

```
config/sample_interval
```

このプロパティーは、zonestatd サンプリング間隔を設定します。これは、ゾーン監視デーモン zonestatd がリソースの使用状況のサンプリングに使用する間隔です。また、プロセッサセットの変更、リソース制御の変更、ゾーン状態の変更などの構成変更を判定するためにも使用されます。

デフォルトの間隔は 5 秒です。

zonestat サービスは、拡張アカウンティング機能を使用します。まだ有効になっていない場合は、プロセスアカウンティングリソースの追跡が有効になり、プロセスアカウンティングファイルが構成されます。zonestat サービスは、その構成された間隔でプロセスアカウンティングログを出力します (zonestat(1) を参照)。

拡張プロセスアカウンティングが外部で有効になっている場合、zonestat サービスは、構成に従ってそのプロセスアカウンティングログを使用します。アカウンティングログは出力されませんが、外部でアカウンティングログが出力されている場合には、このサービスは正しく動作します。

名前

zpool – ZFS ストレージプールの構成

形式

```
zpool [-?]

zpool help command | help | property property-name

zpool add [-f] [-n [-l]] pool vdev ...

zpool attach [-f] pool device new_device

zpool clear [-F [-n]] pool [device]

zpool create [-f] [-n [-l]] [-o property=value] ...
     [-O file-system-property=value] ... [-m mountpoint]
     [-R root] pool vdev ...

zpool destroy [-f] pool

zpool detach pool device

zpool export [-f] pool ...

zpool get "all" | property[,...] pool ...

zpool history [-il] [pool] ...

zpool import [-d path ...] [-D]

zpool import [-o mntopts] [-o property=value] ... [-d path ... |
     -c cachefile] [-D] [-f] [-m] [-N] [-R root] [-F [-n [-l]]] -a

zpool import [-o mntopts] [-o property=value] ... [-d path ... |
     -c cachefile] [-D] [-f] [-m] [-R root] [-F [-n [-l]]]
     pool | id [newpool]

zpool iostat [-T d|u ] [-v [-l]] [pool] ... [interval[count]]

zpool list [-H] [-o property[,...]] [-T d|u ] [pool] ... [interval[count]]

zpool offline [-t] pool device ...

zpool online [-e] pool device ...

zpool remove pool device ...

zpool replace [-f] pool device [new_device]

zpool scrub [-s] pool ...

zpool set property=value pool

zpool split [-R altroot] [-n [-l]] [-o mntopts] [-o property=value] pool
     newpool [device ...]

zpool status [-l] [-v] [-x] [-T d|u ] [pool] ... [interval[count]]

zpool upgrade

zpool upgrade -v

zpool upgrade [-V version] -a | pool ...
```

機能説明

zpool コマンドは、ZFS ストレージプールを構成します。ストレージプールは、ZFS データセット用の物理ストレージおよびデータ複製を提供するデバイスの集合体です。

ストレージプール内部のすべてのデータセットは、同一の領域を共有します。データセットの管理については、zfs(1M) を参照してください。

仮想デバイス (vdev)

「仮想デバイス」とは、特定のパフォーマンスおよび障害特性に従って編成された単一のデバイスまたはデバイスの集合体を指します。次の仮想デバイスがサポートされています。

disk
: ブロックデバイス。通常は /dev/dsk 内に存在します。ZFS では、個別のスライスまたはパーティションを使用することが可能です。ただし、推奨されているのは、ディスク全体を使用する操作モードです。ディスクは、フルパスで指定することも、短縮名 (「/dev/dsk」以下のパスの相対部分) を使用することもできます。スライスやパーティションの指定を省略すると、ディスク全体を指定できます。また、ディスクの現在の場所を指定する /dev/chassis/.../disk パスを使って、ディスク全体を指定することもできます。ディスク全体を指定すると、ZFS によりディスクのラベルが必要に応じて自動的に設定されます。

file
: 通常のファイル。ファイルをバッキングストアとして使用することは決してお勧めできません。ファイルのフォルトトレランスは、それが存在するファイルシステムと同程度に過ぎません。これは、主に実験用途を念頭に置いて設計されています。ファイルは、フルパスで指定する必要があります。

mirror
: 2 つ以上のデバイスのミラー。データの複製方法は、ミラーのすべてのコンポーネントで同じです。サイズ *X* のディスク *N* 台で構成されるミラーは、*X* バイトの格納が可能であり、(*N-1*) 台のデバイスで障害が発生してもデータの整合性が保たれます。

raidz
raidz1
raidz2
raidz3
: RAID-5 の一種。これを使用すると、パリティー分配の向上、および「RAID-5 書き込みホール」(電力喪失によりデータおよびパリティーに矛盾が発生する) の除去が可能になります。データおよびパリティーは、raidz グループ内の全ディスク間でストライプ化されます。

  raidz グループは、シングルパリティー、ダブルパリティー、トリプルパリティーのいずれかを持つことができます。これはそれぞれ、raidz グループがデータを一切失うことなく、1 つ、2 つ、または 3 つの障害に耐えられることを意味します。raidz1 vdev タイプはシングルパリティー raidz グループを、raidz2

vdev タイプはダブルパリティー raidz グループを、raidz3 vdev タイプはトリプルパリティー raidz グループを指定します。raidz vdev タイプは raidz1 の別名です。

サイズ *X* のディスク *N* 台 (パリティーディスク *P* 台を含む) で構成される raidz グループには、約 (*N-P*)*X バイトを格納可能です。また、*P* 台のデバイスに障害が発生してもデータの整合性は損なわれません。raidz グループ内のデバイスの最小数は、パリティーディスクの数に 1 を加えたものです。パフォーマンス向上のために推奨される数は、3 から 9 までです。

`spare`
特定のプールで使用可能なホットスペアを追跡する特殊な擬似 vdev。詳細は、「ホットスペア」節を参照してください。

`log`
別個のインテントログデバイス。複数のログデバイスを指定すると、書き込みの負荷がデバイス間で分散されます。ログデバイスはミラー化できます。ただし、インテントログで raidz vdev タイプはサポートされていません。詳細は、「インテントログ」の節を参照してください。

`cache`
ストレージプールのデータをキャッシュするために使用するデバイス。キャッシュデバイスをミラーまたは `raidz` グループとして構成することはできません。詳細は、「キャッシュデバイス」の節を参照してください。

仮想デバイスを入れ子にすることはできません。このため、ミラーまたは raidz 仮想デバイスに含めることができるのは、ファイルまたはディスクだけです。ミラーをミラー (またはその他の組み合わせを使用) することはできません。

プールでは、構成の最上位に任意の数の仮想デバイス (「ルート vdevs」と呼ばれる) を含めることができます。最上位の全デバイス間でデータが動的に配分されることで、デバイス間のデータ均衡が保たれます。新しい仮想デバイスが追加されると、ZFS は使用可能になった新規デバイスにデータを自動的に格納します。

コマンド行で、複数の仮想デバイスを 1 つずつスペースで区切って指定できます。キーワード「mirror」や「raidz」は、グループが終了し、別のグループが開始することを示す場合に使用します。たとえば、次のコマンドは、それぞれが 2 台のディスクのミラーである 2 つのルート vdevs を作成します。

```
# zpool create mypool mirror c0t0d0 c0t1d0 mirror c1t0d0 c1t1d0
```

また、次のコマンドも使用可能です。

```
# zpool create tank \
mirror \
    /dev/chassis/RACK29.U01-04/DISK_00/disk \
    /dev/chassis/RACK29.U05-08/DISK_00/disk \
mirror \
    /dev/chassis/RACK29.U01-04/DISK_01/disk \
    /dev/chassis/RACK29.U05-08/DISK_01/disk
```

デバイスの障害と復旧

ZFS では、デバイス障害およびデータ破壊を処理するための豊富なメカニズムがサポートされています。すべてのメタデータおよびデータのチェックサムが実行され、破壊が検出されると、ZFS により不正なデータが正常なコピーを使って自動的に修復されます。

これらの機能を活用するため、プールでミラー化グループまたは raidz グループを使って、なんらかの形式の冗長性を利用する必要があります。ZFS では、非冗長構成 (各ルート vdev は単なるディスクまたはファイル) での実行がサポートされていますが、この方法は決してお勧めできません。ビット破壊が 1 回発生するだけで、データの一部またはすべてが使用不可になる可能性があるためです。

プールの健全性ステータスは、次の 3 つのステータスのいずれかで表されます。 オンライン、機能低下、または障害ステータスです。オンラインプールでは、すべてのデバイスが正常に動作しています。機能低下プールでは、1 つ以上のデバイスで障害が発生していますが、冗長性構成のためにデータを引き続き使用できます。障害が発生したプールでは、メタデータが壊れているか、1 つ以上のデバイスで障害が発生しており、動作を継続するための複製が不足しています。

ミラーや raidz デバイスなどの最上位 vdev の健全性は、それに関連付けられた vdev (コンポーネントデバイス) の状態に影響される可能性があります。最上位 vdev とコンポーネントデバイスの状態は、次のいずれかになります。

`DEGRADED`

1 つ以上のコンポーネントデバイスがオフラインであるため、1 つ以上の最上位 vdev が機能低下状態になっています。動作を継続するための十分な複製が存在します。

1 つ以上のコンポーネントデバイスが機能低下または障害状態になっていますが、動作を継続するための十分な複製が存在します。その元となる条件は次のとおりです。

- チェックサムエラーの数が許容レベルを超え、デバイスの機能低下が発生してなんらかの障害の可能性を示しています。必要に応じて、ZFS はそのデバイスを引き続き使用します。
- 入出力エラーの数が許容レベルを超えています。動作を継続するための複製が不足しているため、デバイスを障害状態としてマークできませんでした。

`FAULTED`

1 つ以上のコンポーネントデバイスがオフラインであるため、1 つ以上の最上位 vdev が障害状態になっています。動作を継続するための複製が不足しています。

1 つ以上のコンポーネントデバイスが障害状態になっており、動作を継続するための複製が不足しています。その元となる条件は次のとおりです。

- デバイスを開くことはできましたが、内容が期待された値と一致しませんでした。

- 入出力エラーの数が許容レベルを超え、デバイスは障害発生のため、今後使用できなくなりました。

OFFLINE
: デバイスは「zpool offline」コマンドによって明示的にオフラインになりました。

ONLINE
: デバイスはオンライン状態で動作しています。

REMOVED
: システムの稼働中にデバイスが物理的に取り外されました。デバイスの取り外しの検出はハードウェアに依存しており、一部のプラットフォームではサポートされていない場合があります。

UNAVAIL
: デバイスがオープンできないデバイスが使用不可状態のときにプールをインポートすると、デバイスのパスは最初から間違っているため、デバイスはパスの代わりに一意の識別子によって識別されます。

デバイスを削除したあとでシステムに再度接続すると、ZFS は自動的にデバイスをオンラインにしようと試みます。デバイスの接続の検出はハードウェアに依存しており、一部のプラットフォームではサポートされていない場合があります。

ホットスペア

ZFS では、デバイスを「ホットスペア」としてプールに関連付けることができます。これらのデバイスはプール内でアクティブには使用されませんが、特定のアクティブデバイスで障害が発生すると、そのデバイスは特定のホットスペアで自動的に置き換えられます。ホットスペアを含むプールを作成するには、「spare」 vdev と任意の数のデバイスを指定します。たとえば、次のように指定します。

```
# zpool create pool mirror c0d0 c1d0 spare c2d0 c3d0
```

スペアは zpool add コマンドで追加し、zpool remove コマンドで削除することができます。スペア置換が開始されると新しい「spare」vdev が構成内に作成されますが、元のデバイスが交換されるとそのスペアは削除されます。この時点で、別のデバイスで障害が発生すると、そのホットスペアは再度使用可能となります。

処理中のスペア置換を取り消すには、そのホットスペアを切り離します。障害の発生した元のデバイスが切り離されると、そのデバイスが構成内で占めていた位置にホットスペアが配置され、そのホットスペアがすべてのアクティブプールのスペアリストから削除されます。

障害が発生した元のデバイスが物理的に置き換えられてオンラインに戻るか、またはエラーが FMA イベントを通して、あるいは zpool online コマンドまたは zpool clear コマンドを使用してクリアーされ、かつ元のデバイスの状態が正常になった場合は、INUSE スペアデバイスが再度 AVAIL になります。

ログデバイスをスペアで置き換えることはできません。

インテントログ

ZFS インテントログ (ZIL) は、同期トランザクションの POSIX 要件を満たしています。たとえば、多くの場合、データベースがシステムコールから戻るときは、そのトランザクションが安定したストレージデバイス上に置かれている必要があります。NFS やその他のアプリケーションでは、データの安定性を確保するために `fsync()` も使用できます。デフォルトでは、インテントログはメインプール内のブロックから割り当てられます。ただし、NVRAM や専用ディスクなどの別個のインテントログデバイスを使用することにより、パフォーマンスを向上できる可能性があります。例:

```
# zpool create pool c0d0 c1d0 log c2d0
```

複数のログデバイスを指定することもでき、それらをミラー化することもできます。複数のログデバイスをミラー化する例については、「使用例」の節を参照してください。

ログデバイスは、より大規模なプールの一部として、追加、置き換え、接続、切り離し、インポート、およびエクスポートできます。ミラー化されたログデバイスは、ログの最上位レベルのミラーを指定することによって削除できます。

キャッシュデバイス

ストレージプールに「キャッシュデバイス」としてデバイスを追加できます。これらのデバイスにより、メインメモリーとディスクの間にキャッシュレイヤーが追加されます。集中的に読み取りを行う作業負荷の場合、作業セットのサイズがメインメモリーにキャッシュできるサイズを大きく超えますが、キャッシュデバイスを使用すると、この作業セットの大部分を低遅延メディアから提供できるようになります。キャッシュデバイスを使用すると、ほぼ静的なコンテンツをランダムに読み込む作業負荷のパフォーマンスが大幅に向上します。

キャッシュデバイスを含むプールを作成するには、「cache」`vdev` と任意の数のデバイスを指定します。例:

```
# zpool create pool c0d0 c1d0 cache c2d0 c3d0
```

キャッシュデバイスは、ミラー化することも、raidz 構成に含めることもできません。キャッシュデバイスで読み取りエラーが検出されると、ミラー化構成または raidz 構成に含まれている可能性があるオリジナルのストレージプールデバイスに対して、その読み取り I/O が再発行されます。

キャッシュデバイスの内容は、他のシステムキャッシュによるキャッシュと同様、揮発性であるとみなされます。

プロセス数

インポートされたそれぞれのプールには、`zpool-`*poolname* という名の関連プロセスがあります。このプロセスのスレッドはプールの入出力処理スレッドで、プールに関連付けられたすべての入出力に対して、圧縮、チェックサム、およびそのほかのタスクを処理します。このプロセスは、システムのストレージプールの CPU 使用状況を確認するために用意されています。このプロセスがあるため、インタフェースは不安定になっています。

プロパティー

各プールには、いくつかのプロパティーが関連付けられています。一部のプロパティーは読み取り専用の統計ですが、その他はプールの動作を変更する構成可能なプロパティーです。次のプロパティーは読み取り専用です。

`allocated`
物理的に割り当てられているプール内の記憶容量。このプロパティーは、列の短縮名 `alloc` で参照することもできます。

`capacity`
プール内で使用されている領域の割合。このプロパティーは、列名の短縮形「cap」を使用しても参照できます。

`dedupratio`
プールに対して指定された複製解除率。乗数で表記されます。この値は単一の 10 進数として表記されます。たとえば、`dedupratio` 値が `1.76` の場合、1.76 単位のデータが格納されていたものの、実際には 1 単位のディスク容量だけが消費されていたことを示します。このプロパティーは、列の短縮名 `dedup` を使用して参照することもできます。

複製解除を有効にするには、次のように指定します。

```
# zfs set dedup=on pool/dataset
```

デフォルト値は `off` です。

複製解除機能についての詳細は、`zfs(1M)` を参照してください。

`free`
プール内の割り当てられないブロック数。

`guid`
プールの一意の識別子。

`health`
プールの現在の健全性。健全性は、「`ONLINE`」、「`DEGRADED`」、「`FAULTED`」のいずれかです。

`size`
ストレージプールの合計サイズ。

これらの領域使用状況プロパティーでは、ストレージプールで使用可能な実際の物理容量が報告されます。物理容量は、内部の全データセットが実際に使用可能な総容量とは異なる場合があります。raidz 構成で使用される容量は、書き込まれるデータの特性により異なります。また、ZFS は、`zfs(1M)` コマンドが考慮に入れる内部アカウンティングのための容量を確保します。しかし、`zpool` コマンドでは内部アカウンティングを考慮しません。容量に余裕のある適切なサイズのプールの場合、これらの影響は見えません。小さなプールまたは容量全体がほぼ使用されているプールの場合、これらの相違はより目立つようになります。

次のプロパティーは、作成時およびインポート時に設定できます。

altroot
代替ルートディレクトリ。設定されている場合、プール内のすべてのマウントポイントの先頭にこのディレクトリが付加されます。このプロパティーは、マウントポイントが信頼できない不明なプールを調べるときや、通常のパスが有効でない代替ブート環境で使用できます。altroot は永続的プロパティーではありません。システムが稼働しているときにのみ有効です。altroot を設定すると、デフォルトで cachefile=none が使用されます。ただし、これは明示的な設定を使用して上書きできます。

次のプロパティーは、インポート時に設定できます。

readonly=on | off
このプールを変更できるかどうかを制御します。有効になっている場合は、そのプールが読み取り/書き込みモードでインポートされるまで、インテントログ内にのみ存在するどの同期データにもアクセスできません。

読み取り専用モードでのプールのインポートには、次の制限事項があります。

- インポート中に追加のプールプロパティーを設定しようとしても無視されます。
- ファイルシステムのマウントはすべて、読み取り専用 (ro) マウントオプションを含むように変換されます。

読み取り専用モードでインポートされたプールは、そのプールをエクスポートおよびインポートすることによって読み取り/書き込みモードに戻すことができます。

次のプロパティーは作成時とインポート時に設定でき、あとで zpool set コマンドを使用して変更できます。

autoexpand=on | off
基本となる LUN が増えたときに、自動プール拡張を制御します。on に設定すると、拡張されたデバイスのサイズに応じてプールのサイズが変更されます。デバイスがミラーまたは raidz に含まれている場合、プールで新しい容量を使用できるようにするには、そのミラー/raidz グループ内のすべてのデバイスを拡張する必要があります。デフォルトの動作は off です。このプロパティーは、列の短縮名 expand を使用して参照することもできます。

autoreplace=on | off
自動デバイス交換を制御します。「off」に設定されている場合、管理者が「zpool replace」コマンドを使ってデバイス交換を開始する必要があります。「on」に設定されている場合、そのプールに以前属していたデバイスと物理的に同じ位置にある新しいデバイスは、いずれも自動的にフォーマットされ、置き換えられます。デフォルトの動作は「off」です。このプロパティーは、列名の短縮形「replace」を使用しても参照できます。

`bootfs=`*pool*/*dataset*
ルートプールのデフォルトのブート可能データセットを示します。このプロパティーは、主にインストールプログラムとアップグレードプログラムによって設定されるものと想定されています。

`cachefile=`*path*`|none`
プール構成をキャッシュする場所を制御します。システムの起動時にすべてのプールを見つけるには、ルートファイルシステムに保存されている構成データのキャッシュされたコピーが必要です。システムのブート時に、このキャッシュ内のすべてのプールが自動的にインポートされます。インストール環境やクラスタ化環境などの一部の環境では、プールが自動的にインポートされないようにするために、この情報を別の場所にキャッシュする必要があります。このプロパティーを設定すると、プール構成を別の場所にキャッシュし、あとで「`zpool import -c`」コマンドを使用してインポートできます。特殊な値「`none`」に設定すると、キャッシュされることのない一時プールが作成されます。特殊な値 `''` (空の文字列) に設定すると、デフォルトの場所が使用されます。

同じキャッシュファイルを複数のプールで共有できます。プールの追加時と削除時に、このファイルはカーネルによって破棄され再作成されるため、このファイルにアクセスしようとするときは注意が必要です。キャッシュファイルを使用している最後のプールがエクスポートまたは破棄されると、そのファイルは削除されます。

`dedupditto=`*number*
コピー数のしきい値を設定します。複製解除されたブロックの参照数がこのしきい値を超えると、そのブロックの別の複製コピーが自動的に格納されます。デフォルト値は `0` です。

`delegation=on|off`
データセットに定義されているデータセットのアクセス権に基づいて非特権ユーザーにアクセス権を付与するかどうかを制御します。ZFS 委任管理の詳細については、`zfs(1M)` を参照してください。

`failmode=wait|continue|panic`
壊滅的なプール障害が発生した場合のシステムの動作を制御します。通常は、配下のストレージデバイスへの接続が失われた場合や、プール内のすべてのデバイスに障害が発生した場合に、このような状況になります。このような場合の動作は、次のように決定されます。

`wait`
デバイスへの接続が復元され、エラーがクリアされるまで、プールに対するすべての入出力アクセスをブロックします。デバイスの問題が解決されるまで、プールの状態は wait のままです。これはデフォルトの動作です。

`continue`
新しい書き込み入出力要求には EIO を返しますが、正常な残りのデバイスに対する読み取りは許可します。まだディスクにコミットされていない書き込み要求はブロックされます。

panic
コンソールにメッセージを出力し、システムクラッシュダンプを生成します。

listsnaps=on | off
-t オプションを指定せずに「zfs list」を実行しているときに、このプールに関連付けられたスナップショットに関する情報を出力するかどうかを制御します。デフォルト値は「off」です。

version=*version*
プールの現在のディスク上バージョン。この値を増やすことはできても、減らすことはできません。プールを更新する方法としては zpool upgrade コマンドをお勧めしますが、下位互換性のために特定のバージョンが必要な場合には、このプロパティーを使用できます。このプロパティーには、1 から zpool upgrade -v で報告される現在のバージョンまでの任意の数値を設定できます。

サブコマンド

状態を変更するサブコマンドはすべて、元の形式でプールに永続的に記録されます。

zpool コマンドの提供するサブコマンドを使用すると、ストレージプールの作成と破棄、ストレージプールへの容量の追加、およびストレージプールに関する情報提供を行えます。サポートされているサブコマンドは次のとおりです。

zpool -?
ヘルプメッセージを表示します。

zpool help *command* | help | *property property-name*
zpool コマンドの使用法を表示します。 特定のコマンドまたはプロパティーのヘルプを表示できます。 特定のコマンドまたはプロパティーのヘルプを表示すると、コマンドの構文または使用可能なプロパティー値が表示されます。引数を指定しないで zpool help を使用すると、zpool コマンドの完全なリストが表示されます。

zpool add [-f] [-n [-l]]*pool vdev* ...
指定された仮想デバイスを指定したプールに追加します。*vdev* の仕様については、「仮想デバイス」の節を参照してください。-f オプションの動作、および実行されるデバイス検査については、「zpool create」サブコマンドを参照してください。

-f
使用中と表示されていたり、競合する複製レベルが指定されていたとしても、vdev の使用を強制します。この方法で、すべてのデバイスを上書きできるわけではありません。

-n
vdev を実際に追加することなく、使用される構成を表示します。実際のプール作成は、特権の不足またはデバイス共有のために失敗する可能性があります。

`-l`
可能な場合、`-n` は現在の `/dev/chassis` の場所の形式で構成を表示します。

現在構成されているディスクは、定足数デバイスとして zpool に追加しないでください。ディスクをプールに追加したあとで、そのディスクを定足数デバイスとして構成できます。

`zpool attach` [`-f`] *pool device new_device*
new_device を既存の `zpool` デバイスに接続します。既存のデバイスを、raidz 構成の一部にすることはできません。*device* が現在ミラー構成の一部ではない場合、*device* は *device* および *new_device* の 2 方向のミラーに自動的に変換されます。*device* が 2 方向ミラーの一部である場合、*new_device* を接続すると 3 方向ミラーが作成され、以下同様に作成されます。どの場合でも、*new_device* の再同期化がすぐに開始されます。

`-f`
new_device が使用中と表示されている場合でも、これを強制的に使用します。この方法で、すべてのデバイスを上書きできるわけではありません。

`zpool clear` [`-F` [`-n`]] *pool* [*device*] ...
プール内のデバイスエラーをクリアします。引数を指定しない場合、プール内のすべてのデバイスエラーがクリアされます。1 つ以上のデバイスを指定すると、指定したデバイスに関連するエラーだけがクリアされます。

`-F`
開くことのできないプールに対して回復モードを開始します。プール内にある最後のいくつかのトランザクションを破棄して、開くことのできる状態に戻そうと試みます。このオプションを使用しても、損傷を受けたすべてのプールを回復できるわけではありません。成功した場合、破棄されたトランザクションのデータは回復されずに失われます。

`-n`
`-F` フラグと組み合わせて使用します。トランザクションの破棄によってプールが開くことのできる状態になるかどうかを調べますが、実際にトランザクションを破棄することはありません。

`zpool create` [`-f`] [`-n` [`-l`]] [`-o` *property=value*] ... [`-O` *file-system-property=value*] ... [`-m` *mountpoint*] [`-R` *root*] *pool vdev* ...
コマンド行で指定した仮想デバイスを含む新規ストレージプールを作成します。プール名の先頭は文字でなければなりません。また、プール名に含めることができるのは、英数字、下線 (「_」)、ダッシュ (「-」)、およびピリオド (「.」) です。`mirror`、`raidz`、`spare`、`log` の各プール名、およびパターン `c[0-9]` で始まる名前は予約されています。`vdev` の仕様については、「仮想デバイス」の節を参照してください。

このコマンドは、指定された各デバイスがアクセス可能であり、現在別のサブシステムにより使用されていないことを確認します。ある種の用法 (現在マウントされている、専用のダンプデバイスとして指定されているなど) では、ZFS によ

るデバイスの使用が妨げられます。-f オプションを使用すると、その他の用法 (既存の UFS ファイルシステムの使用など) を上書きできます。

このコマンドは、プールの複製方法が一貫しているかどうかもチェックします。-f を指定せずに、単一のプール内で冗長ストレージと非冗長ストレージの結合を試みたり、ディスクとファイルの混在を試みると、エラーが発生します。-f を指定せずに、単一の raidz またはミラーグループ内でサイズの異なるデバイスを使用しても、エラーのフラグが付けられます。

-R オプションを指定しない場合の、デフォルトのマウントポイントは「/*pool*」です。このマウントポイントは存在しないか、空でなければなりません。さもないと、ルートデータセットをマウントすることはできません。これは、-m オプションを使って上書きできます。

-f
: 使用中と表示されていたり、競合する複製レベルが指定されていたとしても、vdev の使用を強制します。この方法で、すべてのデバイスを上書きできるわけではありません。

-n
: 使用する設定を、実際にプールを作成せずに表示します。実際のプール作成は、特権不足やデバイスが現在使用中などのために、まだ失敗する可能性があります。

-l
: 可能な場合、-n は現在の /dev/chassis の場所の形式で構成を表示します。

-o *property=value* [-o *property=value*] ...
: 指定されたプールのプロパティーを設定します。設定できる有効なプロパティーのリストについては、「プロパティー」の節を参照してください。

-O *file-system-property=value*
[-O *file-system-property=value*] ...
: プールの最上位レベルのファイルシステムの指定されたプロパティーを設定します。設定できる有効なプロパティーのリストについては、zfs(1M) の「プロパティー」の節を参照してください。

-R *root*
: 「-o cachefile=none,altroot=*root*」と同等

-m *mountpoint*
: プールの最上位レベルのファイルシステムのマウントポイントを設定します。デフォルトのマウントポイントは、「/*pool*」です。または、altroot が指定されている場合は「altroot/*pool*」です。このマウントポイントは絶対パス、「legacy」または「none」でなければなりません。データセットマウントポイントの詳細は、zfs(1M)を参照してください。

`zpool destroy [-f] pool`
指定したプールを破棄し、すべてのデバイスを解放してほかの用途で使用できるようにします。このコマンドは、プールを破棄する前に、アクティブなデータセットすべてのマウント解除を試みます。

`-f`
プール内に含まれるアクティブなデータセットすべてのマウント解除を強制的に行います。

`zpool detach pool device`
ミラー化されたストレージプールからデバイスまたはスペアを切り離します。既存のデバイスが物理的に置き換えられた場合は、RAID-Z ストレージプールからスペアを切り離すこともできます。また、スペアで置き換えられた場合は、RAID-Z ストレージプールの既存デバイスを切り離すこともできます。データの有効な複製がほかに存在しない場合、この操作は拒否されます。

`zpool export [-f] pool ...`
指定したプールをシステムからエクスポートします。すべてのデバイスにエクスポート済みのマークが付けられますが、別のサブシステムからは引き続き使用中と見なされます。十分な数のデバイスが存在するかぎり、このデバイスをシステム間 (エンディアンの異なるシステムを含む) で移動またはインポートすることが可能です。

プールをエクスポートする前に、プール内のすべてのデータセットがマウント解除されます。

プールを可搬性のあるものにするため、`zpool` コマンドに単なるスライスではなく、ディスク全体を指定する必要があります。これにより、ZFS は可搬性のある EFI ラベルをディスクに設定します。そうしなければ、エンディアンの異なるプラットフォーム上のディスクドライバは、ディスクを認識しません。

`-f`
「`unmount -f`」コマンドを使用して、すべてのデータセットを強制的にマウント解除します。

このコマンドは強制的にプールをエクスポートします。

`zpool get "all" | property[,...] pool ...`
指定されたストレージプールについて、指定されたプロパティー (「`all`」が指定された場合はすべてのプロパティー) のリストを取得します。これらのプロパティーは、次のフィールドで表示されます。

```
name          Name of storage pool
 property      Property name
 value         Property value
 source        Property source, either 'default' or 'local'.
```

使用可能なプールプロパティーの詳細については、「プロパティー」の節を参照してください。

zpool history [-il] [*pool*] ...

指定したプール (プールを指定しない場合はすべてのプール) のコマンド履歴を表示します。

-i

ユーザーによって開始されたイベントに加え、内部で記録された ZFS イベントを表示します。

-l

ログレコードを長形式で表示します。これには、標準形式のほかに、ユーザー名、ホスト名、および操作が実行されたゾーンが含まれます。

zpool import [-d *path* ... | -c *cachefile*] [-D]

インポートに使用可能なプールを一覧表示します。-d オプションを指定しない場合、このコマンドは /dev/dsk 内のデバイスを検索します。-d オプションは、複数回指定できます。このオプションを指定すると、すべてのディレクトリおよびデバイスパスが検索されます。デバイスがエクスポートされたプールの一部として表示される場合、このコマンドは、プールの名前、数値識別子、および各デバイスやファイルの *vdev* レイアウトと現在の健全性を含むプールの要約を表示します。-D オプションを指定しないかぎり、破棄プール、つまり zpool destroy コマンドを使用して以前に破棄されたプールは表示されません。

数値識別子は一意であり、エクスポートされた同名のプールを複数利用可能な場合にプール名の代わりに使用できます。

-c *cachefile*

「cachefile」プールプロパティーで作成された指定の cachefile から構成を読み取ります。デバイスを検索する代わりに、この cachefile が使用されます。

-d *path*

path 内のデバイスまたはファイルを検索します。ここで、*path* にはディレクトリまたはデバイスパスを指定できます。-d オプションは複数回指定できます。

-D

破棄されたプールだけを一覧表示します。

zpool import [-o *mntopts*] [-o *property*=*value*] ... [-d *path* ... | -c *cachefile*] [-D] [-f] [-m] [-N] [-R *root*] [-F [-n [-l]]] -a

検索ディレクトリまたはデバイスパスで見つかったすべてのプールをインポートします。十分な数のデバイスを使用可能なプールがすべてインポートされることを除き、前のコマンドと同じです。-D オプションを指定しないかぎり、破棄プール、つまり、zpool destroy コマンドを使用して以前に破棄されたプールはインポートされません。

-o *mntopts*
プール内のデータセットのマウント時に使用するマウントオプションのコンマ区切りのリスト。データセットのプロパティーとマウントオプションの説明については、zfs(1M) を参照してください。

-o *property=value*
インポートされたプールに、指定されたプロパティーを設定します。使用可能なプールプロパティーの詳細については、「プロパティー」の節を参照してください。

-c *cachefile*
「cachefile」プールプロパティーで作成された指定の cachefile から構成を読み取ります。デバイスを検索する代わりに、この cachefile が使用されます。

-d *path*
デバイスまたはファイルを *path* 内で検索します。-d オプションは複数回指定できます。このオプションは -c オプションと同時には使用できません。

-D
破棄されたプールだけをインポートします。-f オプションも指定する必要があります。

-f
プールが潜在的にアクティブであると表示されている場合でも、インポートを強制的に実行します。

-F
インポート不可能なプールの回復モード。最後のいくつかのトランザクションを破棄することによって、プールをインポート可能な状態に戻そうとします。このオプションを使用しても、損傷を受けたすべてのプールを回復できるわけではありません。成功した場合、破棄されたトランザクションのデータは回復されずに失われます。このオプションは、プールがインポート可能であるか、すでにインポートされている場合には無視されます。

-a
プールを検索し、見つかったすべてのプールをインポートします。

-m
ログデバイスがない場合のプールのインポートを許可します。

-R *root*
「cachefile」プロパティーを「none」、「*altroot*」プロパティーを「*root*」に設定します。

-N
ファイルシステムをマウントせずにプールをインポートします。

-n
: -F 回復オプションとともに使用します。インポート不可能なプールを再度インポート可能にできるかどうかを判断しますが、実際にはプール回復を実行することはありません。プール回復モードの詳細については、前述の -F オプションを参照してください。

-l
: 可能な場合、-n は現在の /dev/chassis の場所の形式で情報を表示します。

zpool import [-o *mntopts*] [-o *property=value*] ... [-d *path* ... | -c *cachefile*] [-D] [-f] [-m] [-R *root*] [-F [-n [-l]]] *pool* | *id* [*newpool*]
: 特定のプールをインポートします。プールは、名前または数値識別子を使って識別できます。*newpool* が指定されている場合、プールは永続的な名前 *newpool* を使用してインポートされます。それ以外の場合は、エクスポートされた名前と同じ名前を使ってインポートが行われます。ルートプールを新しい名前でインポートしないでください。そうでないと、システムがブートしなくなる可能性があります。

zpool export を最初に実行せずにシステムからデバイスを削除すると、そのデバイスは潜在的にアクティブな状態として表示されます。これがエクスポートの失敗を示すのか、デバイスが別のホストにより実際に使用されていることを示すのかを見分けることはできません。この状態のプールをインポートするには、-f オプションを指定する必要があります。

-o *mntopts*
: プール内のデータセットのマウント時に使用するマウントオプションのコンマ区切りのリスト。データセットのプロパティーとマウントオプションの説明については、zfs(1M) を参照してください。

-o *property=value*
: インポートされたプールに、指定されたプロパティーを設定します。使用可能なプールプロパティーの詳細については、「プロパティー」の節を参照してください。

-c *cachefile*
: 「cachefile」プールプロパティーで作成された指定の cachefile から構成を読み取ります。デバイスを検索する代わりに、この cachefile が使用されます。

-d *path*
: デバイスまたはファイルを *path* 内で検索します。-d オプションは複数回指定できます。このオプションは -c オプションと同時には使用できません。

-D
: 破棄されたプールをインポートします。-f オプションも指定する必要があります。

-f
プールが潜在的にアクティブであると表示されている場合でも、インポートを強制的に実行します。

-F
インポート不可能なプールの回復モード。最後のいくつかのトランザクションを破棄することによって、プールをインポート可能な状態に戻そうとします。このオプションを使用しても、損傷を受けたすべてのプールを回復できるわけではありません。成功した場合、破棄されたトランザクションのデータは回復されずに失われます。このオプションは、プールがインポート可能であるか、すでにインポートされている場合には無視されます。

-R *root*
「cachefile」プロパティーを「none」、「*altroot*」プロパティーを「*root*」に設定します。

-n
-F 回復オプションとともに使用します。インポート不可能なプールを再度インポート可能にできるかどうかを判断しますが、実際にはプール回復を実行することはありません。プール回復モードの詳細については、前述の -F オプションを参照してください。

-l
可能な場合、-n は現在の /dev/chassis の場所の形式で情報を表示します。

-m
ログデバイスがない場合のプールのインポートを許可します。

zpool iostat [-T d|u] [-v [-l]] [*pool*] ... [*interval*[*count*]]
指定したプールの I/O 統計を表示します。間隔を指定した場合、Ctrl-C キーを押すまで、統計が *interval* 秒ごとに出力されます。*pools* を指定しない場合、システム内のすべてのプールの統計が表示されます。*count* を指定した場合、このコマンドは *count* レポートの出力後に配置されます。

-T d|u
タイムスタンプを表示します。

標準の日付フォーマットに d を指定します。date(1) を参照してください。時刻の内部表現の出力表現に u を指定します。time(2) を参照してください。

-v
詳細な統計。プール全体の統計とともに、プール内の各 *vdevs* の使用状況統計をレポートします。

-l
可能な場合、-v は現在の /dev/chassis の場所の形式で vdev 統計を表示します。

`zpool list` [`-H`] [`-o` *props*[,...]] [`-T` d|u] [*pool*] ...

指定したプールについて、健全性ステータスおよび領域使用状況を一覧表示します。引数を指定しない場合、システム内のすべてのプールが表示されます。

間隔を指定した場合、Ctrl-C キーを押すまで、ステータスおよび領域使用状況が *interval* 秒ごとに表示されます。*count* を指定した場合、このコマンドは *count* レポートの表示後に終了します。

`-H`

スクリプトモード。ヘッダーを表示せず、フィールドを任意のスペース文字ではなく単一のタブで区切ります。

`-o` *props*

表示するプロパティーのコンマ区切りのリスト。有効なプロパティーのリストについては、「プロパティー」の節を参照してください。デフォルトのリストは、`name`、`size`、`allocated`、`free`、`capacity`、`health`、`altroot` です。

`-T` d|u

タイムスタンプを表示します。

標準の日付フォーマットに `d` を指定します。`date(1)` を参照してください。時刻の内部表現の出力表現に `u` を指定します。`time(2)` を参照してください。

`zpool offline` [`-t`] *pool device* ...

指定した物理デバイスをオフラインにします。オフライン状態にある *device* に対して、読み取りや書き込みは行われません。

このコマンドは、スペアまたはキャッシュデバイスには使用できません。

`-t`

一時的に使用する場合に指定します。リブートすると、指定した物理デバイスは以前の状態に戻ります。

`zpool online` [`-e`] *pool device*...

指定した物理デバイスをオンラインにします。

このコマンドは、スペアまたはキャッシュデバイスには使用できません。

`-e`

使用可能なすべての容量を使用するようにデバイスを拡張します。デバイスがミラーまたは `raidz` に含まれている場合、プールで新しい容量を使用できるようにするには、すべてのデバイスを拡張する必要があります。

`zpool remove` *pool device* ...

指定されたデバイスをプールから削除します。現在、このコマンドでは、ホットスペア、キャッシュ、およびログデバイスの削除のみをサポートしています。ミラー化されたログデバイスは、ログの最上位レベルのミラーを指定することによって削除できます。ミラー構成の一部となっている非ログデバイスは、`zpool detach` コマンドを使用すると削除できます。非冗長デバイスおよび raidz デバイスはプールから削除することはできません。

`zpool replace` [`-f`] *pool old_device* [*new_device*]

old_device を *new_device* で置き換えます。これは、*new_device* を接続し、再同期化の実行まで待機してから *old_device* を切り離す操作と同等です。

new_device のサイズは、ミラーまたは raidz 構成内の全デバイスの最小サイズ以上でなければなりません。

new_device は、プールに冗長性がない場合に必要です。*new_device* を指定しない場合、デフォルトの *old_device* が使用されます。この置換形式は、既存のディスクで障害が発生したためにディスクを物理的に交換したあとで実行する場合に役立ちます。この場合、実際には別のディスクであっても、新規ディスクには以前のデバイスと同じ `/dev/dsk` パスが使用されます。これは、ZFS により認識されます。

`zpool status` 出力では、*old_device* はワード `replacing` の下に表示され、文字列 `/old` が付加されます。再同期化が完了すると、`replacing` と *old_device* の両方が自動的に削除されます。再同期化が完了する前に新しいデバイスに障害が発生し、第 3 のデバイスが代わりに取り付けられると、障害が発生した両方のデバイスは `/old` が付加された状態で表示され、再同期化がもう一度開始されます。再同期化が完了すると、両方の `/old` デバイスは、ワード `replacing` とともに削除されます。

`-f`

new_device が使用中と表示されている場合でも、これを強制的に使用します。この方法で、すべてのデバイスを上書きできるわけではありません。

`zpool scrub` [`-s`] *pool* ...

消し込みを開始します。消し込みにより、指定したプール内のすべてのデータが検査され、チェックサムが適正に検証されます。複製された (ミラーまたは raidz) デバイスの場合、消し込み中に検出されたすべての損傷は、ZFS により自動的に修復されます。「`zpool status`」コマンドは、消し込みの進捗状況をレポートします。また、完了時には消し込み結果の概要を出力します。

消し込みおよび再同期化は、非常に類似した操作です。相違点は、再同期化では ZFS が期限切れであることを認識しているデータだけが検査されますが (新規デバイスをミラーに接続する場合や、既存のデバイスを置換する場合など)、消し込みではすべてのデータが検査されるため、ハードウェア障害やディスク障害に起因する非表示のエラーも検出されます。

消し込みと再同期化は、I/O 集約的な操作であるため、ZFS では一度に 1 つの操作だけが許可されます。消し込みがすでに進行している場合、以降の「`zpool scrub`」はエラーを返し、`zpool scrub - s` を使用して現在の消し込みを取り消すことを勧めるアドバイスが表示されます。進行中の再同期化が存在する場合、ZFS は、再同期化が完了するまで書き込みの開始を許可しません。

`-s`

消し込みを停止します。

`zpool set` *property=value pool*
指定されたプールに、特定のプロパティーを設定します。設定可能なプロパティーおよび有効な値については、「プロパティー」の節を参照してください。

`zpool split` [`-R` *altroot*] [`-n` [`-l`]] [`-o` *mntopts*] [`-o` *property=value*] *pool newpool* [*device* ...]
プール内のミラー化された各最上位 vdev から 1 つのディスクを分割し、分割したディスクから新しいプールを作成します。元のプールは 1 つ以上のミラーから構成されている必要があり、再同期化のプロセスが進行中であってはいけません。`split` サブコマンドは、コマンド行でのデバイス指定によって上書きされないかぎり、各ミラー vdev の最後のデバイスを選択します。

device 引数を使用した場合、`split` は指定されたデバイスを新しいプールに含め、どのデバイスも指定されないままであれば、通常どおり、各ミラー vdev 内の最後のデバイスをそのプールに割り当てます。`split` コマンドの結果が不明な場合は、`-n` (「予行演習」) オプションを使用して、コマンドが意図した結果をもたらすことを確認します。

`-R` *altroot*
指定された *altroot* パラメータを新しいプールの代替ルートに使用して、分割後に新しく作成されたプールを自動的にインポートします。前述の「プロパティー」節の `altroot` の説明を参照してください。

`-n`
作成される構成を、実際にプールを分割せずに表示します。実際のプール分割は、特権の不足やデバイスのステータスのために失敗する可能性があります。

`-l`
可能な場合、`-n` は現在の `/dev/chassis` の場所の形式で構成を表示します。

`-o` *mntopts*
プール内のデータセットのマウント時に使用するマウントオプションのコンマ区切りのリスト。データセットのプロパティーとマウントオプションの説明については、`zfs(1M)` を参照してください。`-R` オプションと組み合わせた場合にのみ有効です。

`-o` *property=value*
新しいプールに、指定されたプロパティーを設定します。使用可能なプールプロパティーの詳細については、前述の「プロパティー」の節を参照してください。

`zpool status` [`-l`] [`-v`] [`-x`] [`-T` d|u] [*pool*] ... [*interval*[*count*]]
指定したプールの詳細な健全性ステータスを表示します。*pool* を指定しない場合、システムの各プールのステータスが表示されます。プールおよびデバイスの健全性の詳細については、「デバイスの障害と復旧」の節を参照してください。

間隔を指定した場合、Ctrl-C キーを押すまで、ステータスおよび領域使用状況が *interval* 秒ごとに表示されます。*count* を指定した場合、このコマンドは *count* レポートの表示後に終了します。

消し込みまたは再同期化が進行中の場合、このコマンドは、実行済みの割合 (パーセント) および推定完了時間をレポートします。プール内のデータ量およびシステムのその他のワークロードは変化する場合があるため、これらはどちらも概算値に過ぎません。

`-l`
可能な場合、現在の `/dev/chassis` の場所の形式で vdev 統計を表示します。

`-x`
エラーが発生しているか、使用不可能なプールの状態だけを表示します。

`-v`
詳細なデータエラー情報を表示し、前回のプール消し込みが完了して以降のデータエラーすべての完全なリストを出力します。

`-T d|u`
タイムスタンプを表示します。

標準の日付フォーマットに `d` を指定します。`date(1)` を参照してください。時刻の内部表現の出力表現に `u` を指定します。`time(2)` を参照してください。

`zpool upgrade`
プールのディスク上バージョンを指定します。このバージョンによって、現在実行中のソフトウェアリリースで使用可能なプール機能が決まります。古いプールバージョンを使用し続けることはできますが、一部の機能が使用できなくなる可能性があります。プールは、`zpool upgrade - a` コマンドを使用するとアップグレードできます。以前のソフトウェアバージョンを実行しているシステムで、以降のバージョンのプールにアクセスすることはできません。

`zpool upgrade -v`
現在のソフトウェアでサポートされている ZFS プールバージョンを表示します。現在の ZFS プールバージョンおよびサポートされる以前のバージョンすべてが、各バージョンで提供される機能に関する説明とともに表示されます。

`zpool upgrade [-V` *version*`] -a |` *pool* ...
指定したプールを最新のディスク上バージョンにアップグレードします。このコマンドでプールが古いバージョンであることがわかった場合、続けて `zpool upgrade -a` コマンドを使用すると、そのプールをアップグレードできます。アップグレードされたプールは、以前のソフトウェアリリースを実行するシステムではアクセスできなくなります。

`-a`
すべてのプールをアップグレードします。

`-V` *version*
指定されたバージョンにアップグレードします。現在のバージョンよりも高いバージョンを指定する必要があります。`-V` フラグが指定されていない場合、プールは最新のバージョンにアップグレードされます。

使用例

例 1　RAID-Z ストレージプールを作成する

次のコマンドは、6 台のディスクで構成される単一の raidz ルート *vdev* を含むプールを作成します。

```
# zpool create tank raidz c0t0d0 c0t1d0 c0t2d0 c0t3d0 c0t4d0 c0t5d0
```

例 2　ミラー化されたストレージプールを作成する

次のコマンドは、2 つのミラーを持つプールを作成します。各ミラーには 2 台のディスクが含まれます。

```
# zpool create tank mirror c0t0d0 c0t1d0 mirror c0t2d0 c0t3d0
```

また、ディスクの現在の場所を指定する `/dev/chassis` パスを使って、ディスク全体を指定することもできます。

```
# zpool create tank \
    mirror \
        /dev/chassis/RACK29.U01-04/DISK_00/disk \
        /dev/chassis/RACK29.U05-08/DISK_00/disk \
    mirror \
        /dev/chassis/RACK29.U01-04/DISK_01/disk \
/dev/chassis/RACK29.U05-08/DISK_01/disk
```

例 3　スライスを使って ZFS ストレージプールを作成する

次のコマンドは、2 つのディスクスライスを使ってミラー化されていないプールを作成します。

```
# zpool create tank /dev/dsk/c0t0d0s1 c0t1d0s4
```

また、ディスクの現在の場所を指定する `/dev/chassis` パスを使って、ディスク全体を指定することもできます。

```
# zpool add tank mirror \
        /dev/chassis/RACK29.U01-04/DISK_02/disk \
        /dev/chassis/RACK29.U05-08/DISK_02/disk
```

例 4　ミラーを ZFS ストレージプールに追加する

次のコマンドは、2 つのミラー化ディスクをプール「*tank*」に追加します。ただし、プールは 2 方向ミラーで構成済みであるものとします。プール内のすべてのデータセットは、追加された領域をすぐに利用できます。

```
# zpool add tank mirror c1t0d0 c1t1d0
```

例 5　使用可能な ZFS ストレージプールを一覧表示する

次のコマンドは、システムで使用可能なすべてのプールを一覧表示します。

```
# zpool list
NAME    SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
pool    136G   109M   136G     0%  3.00x  ONLINE  -
```

例 5 使用可能な ZFS ストレージプールを一覧表示する (続き)

```
rpool  67.5G  12.6G  54.9G    18%  1.01x  ONLINE  -
```

例 6 プールのすべてのプロパティーを一覧表示する

次のコマンドは、プールのプロパティーをすべて一覧表示します。

```
% zpool get all pool
NAME  PROPERTY       VALUE       SOURCE
pool  size           464G        -
pool  capacity       0%          -
pool  altroot        -           default
pool  health         ONLINE      -
pool  guid           9893442355858802973  default
pool  version        23          default
pool  bootfs         -           default
pool  delegation     on          default
pool  autoreplace    off         default
pool  cachefile      -           default
pool  failmode       wait        default
pool  listsnapshots  off         default
pool  autoexpand     off         default
pool  dedupditto     0           default
pool  dedupratio     1.00x       -
pool  free           464G        -
pool  allocated      130K        -
```

例 7 ZFS ストレージプールを破棄する

次のコマンドは、プール「*tank*」およびその内部のデータセットをすべて破棄します。

```
# zpool destroy -f tank
```

例 8 ZFS ストレージプールをエクスポートする

次のコマンドは、プール *tank* 内のデバイスをエクスポートします。エクスポートされたデバイスは、再配置したり、あとでインポートしたりすることが可能です。

```
# zpool export tank
```

例 9 ZFS ストレージプールをインポートする

次のコマンドは、使用可能なプールを表示してから、プール「tank」をインポートしてシステムで使用できるようにします。

このコマンドを実行すると、次のような結果が表示されます。

```
# zpool import
  pool: tank
    id: 7678868315469843843
```

例 9 ZFS ストレージプールをインポートする (続き)

```
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

              tank  ONLINE
          mirror-0  ONLINE
            c1t2d0  ONLINE
            c1t3d0  ONLINE

# zpool import tank
```

例 10 すべての ZFS ストレージプールを最新のバージョンにアップグレードする

次のコマンドは、すべての ZFS ストレージプールを最新バージョンのソフトウェアにアップグレードします。

```
# zpool upgrade -a
This system is currently running ZFS pool version 22.

All pools are formatted using this version.
```

例 11 ホットスペアを管理する

次のコマンドは、使用可能なホットスペアを 1 つ含む新しいプールを作成します。

```
# zpool create tank mirror c0t0d0 c0t1d0 spare c0t2d0
```

いずれかのディスクで障害が発生すると、このプールは機能低下状態に陥ります。障害が発生したデバイスを交換するには、次のコマンドを使用します。

```
# zpool replace tank c0t0d0 c0t3d0
```

デバイスが回復されると、スペアは自動的に切り離され、障害が発生した別のデバイスで使用可能となります。ホットスペアをプールから完全に削除するには、次のコマンドを使用します。

```
# zpool remove tank c0t2d0
```

例 12 個別のミラー化されたログデバイスによる ZFS プールの作成

次のコマンドは、2 つの 2 方向ミラーとミラー化されたログデバイスから構成される ZFS ストレージプールを作成します。

```
# zpool create pool mirror c0d0 c1d0 mirror c2d0 c3d0 log mirror \
   c4d0 c5d0
```

例 13 ZFS プールにキャッシュデバイスを追加する

次のコマンドは、キャッシュデバイスとして使用する 2 つのディスクを ZFS ストレージプールに追加します。

例 13 ZFS プールにキャッシュデバイスを追加する (続き)

```
# zpool add pool cache c2d0 c3d0
```

キャッシュデバイスを追加すると、そのキャッシュデバイスにメインメモリーの内容が徐々に書き込まれていきます。キャッシュデバイスのサイズによっては、キャッシュがいっぱいになるまでに 1 時間以上かかる場合もあります。iostat オプションを次のように使用して、容量と読み込みを監視できます。

```
# zpool iostat -v pool 5
```

例 14 ミラー化されたログデバイスを削除する

すぐ下で示す構成を想定した場合、次のコマンドはプール tank 内のミラー化されたログデバイス mirror-2 を削除します。

```
   pool: tank
  state: ONLINE
  scrub: none requested
config:

         NAME        STATE     READ WRITE CKSUM
         tank        ONLINE       0     0     0
           mirror-0  ONLINE       0     0     0
             c6t0d0  ONLINE       0     0     0
             c6t1d0  ONLINE       0     0     0
           mirror-1  ONLINE       0     0     0
             c6t2d0  ONLINE       0     0     0
             c6t3d0  ONLINE       0     0     0
         logs
           mirror-2  ONLINE       0     0     0
             c4t0d0  ONLINE       0     0     0
             c4t1d0  ONLINE       0     0     0
```

```
# zpool remove tank mirror-2
```

例 15 障害が発生した ZFS プールを回復する

プールに障害が発生したが回復可能である場合、このステータスを示すメッセージは、そのプールがキャッシュされていた場合 (前述の cachefile を参照) には zpool status によって表示され、キャッシュされていなかった場合にはそのプールで失敗した zpool import のエラー出力の一部として表示されます。

zpool clear コマンドを使用して、キャッシュされたプールを回復します。

```
# zpool clear -F data
Pool data returned to its state as of Tue Sep 08 13:23:35 2009.
Discarded approximately 29 seconds of transactions.
```

プール構成がキャッシュされなかった場合は、zpool import を回復モードフラグを指定して使用します。

例 15 障害が発生した ZFS プールを回復する (続き)

```
# zpool import -F data
Pool data returned to its state as of Tue Sep 08 13:23:35 2009.
Discarded approximately 29 seconds of transactions.
```

例 16 ログデバイスがない場合の ZFS プールのインポート

次の例は、ログデバイスがない場合にプールのインポートを試みる方法を示しています。-m オプションは、インポート操作を完了するために使用されます。

このプールに追加のデバイスが含まれていることがわかっていますが、その正確な構成は特定できません。

```
# zpool import tank
The devices below are missing, use '-m' to import the pool anyway:
             c5t0d0 [log]

cannot import 'tank': one or more devices is currently unavailable

# zpool import -m tank
# zpool status tank
   pool: tank
  state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas
exist for
         the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
    see: http://www.sun.com/msg/ZFS-8000-2Q
   scan: none requested
config:

         NAME                    STATE     READ WRITE CKSUM
         tank                    DEGRADED     0     0     0
           c7t0d0                ONLINE       0     0     0
         logs
           1693927398582730352   UNAVAIL      0     0     0  was
/dev/dsk/c5t0d0

errors: No known data errors
```

次の例は、ミラー化されたログデバイスがない場合にプールをインポートする方法を示しています。

```
# zpool import tank
The devices below are missing, use ?-m? to import the pool anyway:
mirror-1 [log]
c5t0d0
c5t1d0
```

例 16 ログデバイスがない場合の ZFS プールのインポート (続き)

```
# zpool import -m tank

# zpool status tank
   pool: tank
  state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas
exist for the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
    see: http://www.sun.com/msg/ZFS-8000-2Q
    scan: none requested
config:

         NAME                      STATE     READ WRITE CKSUM
         tank                      DEGRADED     0     0     0
           c7t0d0                  ONLINE       0     0     0
         logs
           mirror-1                UNAVAIL      0     0     0
insufficient replicas
             46385995713041169     UNAVAIL      0     0     0  was
/dev/dsk/c5t0d0
             13821442324672734438  UNAVAIL      0     0     0  was
/dev/dsk/c5t1d0

errors: No known data errors
```

例 17 特定のパスを使用してプールをインポートする

次のコマンドは、プールの特定のデバイスパス (この例では、/dev/dsk/c9t9d9 と /dev/dsk/c9t9d8) を指定することによってプール tank をインポートします。

```
# zpool import -d /dev/dsk/c9t9d9s0 /dev/dsk/c9t9d8s0 tank
```

このプールがディスク全体から構成されている場合でも、特定のデバイスのスライス識別子をコマンドに含める必要があるという既存の制限があります。

終了ステータス

次の終了値が返されます。

0
: 正常終了。

1
: エラーが発生しました。

2
: 無効なコマンド行オプションが指定されました。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	確実

関連項目

zfs(1M), attributes(5)

名前

zstreamdump – zfs send ストリームのデータのフィルタ処理

形式

```
zstreamdump [-C] [-v]
```

機能説明

zstreamdump ユーティリティーは、zfs send コマンドの出力を読み取り、その出力のヘッダーといくつかの統計情報を表示します。zfs(1M) を参照してください。

オプション

次のオプションがサポートされています。

-C
チェックサムの検証を抑制します。

-v
冗長。先頭と末尾のヘッダーだけでなく、すべてのヘッダーをダンプします。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/file-system/zfs
インタフェースの安定性	不確実

関連項目

zfs(1M), attributes(5)