マニュアルページセクション 5: 標準、環境、マクロ

Part No: E37317
2012 年 10 月

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

121210@25097

目次

はじめに 9

紹介 13
Intro(5) 14

Standards, Environments, and Macros 15
acl(5) 16
ad(5) 24
ars(5) 25
ascii(5) 28
attributes(5) 30
audit_binfile(5) 41
audit_flags(5) 43
audit_remote(5) 44
audit_syslog(5) 50
brands(5) 53
cancellation(5) 55
charmap(5) 61
condition(5) 65
crypt_bsdbf(5) 67
crypt_bsdmd5(5) 68
crypt_sha256(5) 69
crypt_sha512(5) 70
crypt_sunmd5(5) 71
crypt_unix(5) 72
datasets(5) 73
device_clean(5) 75

dhcp(5) 77
dhcp_modules(5) 79
environ(5) 80
eqnchar(5) 86
extendedFILE(5) 87
extensions(5) 89
fedfs(5) 90
filesystem(5) 93
fmri(5) 97
fnmatch(5) 120
formats(5) 124
fsattr(5) 129
grub(5) 140
gss_auth_rules(5) 142
hal(5) 143
iconv_1250(5) 144
iconv_1251(5) 150
iconv(5) 159
iconv_646(5) 163
iconv_852(5) 166
iconv_8859-1(5) 173
iconv_8859-2(5) 180
iconv_8859-5(5) 186
iconv_dhn(5) 194
iconv_koi8-r(5) 198
iconv_mac_cyr(5) 206
iconv_maz(5) 214
iconv_pc_cyr(5) 218
iconv_unicode(5) 224
ieee802.11(5) 229
ieee802.3(5) 230
ipfilter(5) 236
isalist(5) 241
kerberos(5) 244
krb5_auth_rules(5) 247
krb5envvar(5) 249

kssl(5) 252
labels(5) 253
largefile(5) 255
ldap(5) 259
lf64(5) 263
lfcompile(5) 270
lfcompile64(5) 273
locale(5) 275
locale_alias(5) 302
man(5) 319
mansun(5) 324
me(5) 328
mech_spnego(5) 333
mm(5) 335
ms(5) 342
mutex(5) 347
mwac(5) 349
nfssec(5) 350
nis(5) 353
nwam(5) 354
openssl(5) 357
pam_allow(5) 361
pam_authtok_check(5) 363
pam_authtok_get(5) 365
pam_authtok_store(5) 367
pam_deny(5) 368
pam_dhkeys(5) 370
pam_dial_auth(5) 372
pam_krb5(5) 373
pam_krb5_migrate(5) 383
pam_ldap(5) 386
pam_list(5) 391
pam_passwd_auth(5) 394
pam_pkcs11(5) 396
pam_rhosts_auth(5) 400
pam_roles(5) 401

pam_sample(5) 404
pam_smbfs_login(5) 406
pam_smb_passwd(5) 408
pam_tsol_account(5) 410
pam_tty_tickets(5) 412
pam_unix_account(5) 414
pam_unix_auth(5) 416
pam_unix_cred(5) 419
pam_unix_session(5) 421
pam_user_policy(5) 422
pam_zfs_key(5) 425
pkcs11_kernel(5) 428
pkcs11_kms(5) 430
pkcs11_softtoken(5) 434
pkcs11_tpm(5) 439
pkg(5) 443
privileges(5) 463
prof(5) 478
rbac(5) 479
regex(5) 482
regexp(5) 491
resource_controls(5) 498
sgml(5) 507
smf(5) 511
smf_bootstrap(5) 521
smf_method(5) 524
smf_restarter(5) 531
smf_security(5) 533
smf_template(5) 537
solaris10(5) 547
solaris(5) 552
standards(5) 558
sticky(5) 565
suri(5) 566
tecla(5) 568
term(5) 590

threads(5) 594
trusted_extensions(5) 601
vgrindefs(5) 602
wbem(5) 605
zones(5) 608

はじめに

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

概要

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

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

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

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

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

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

  このセクションでは、次の特殊文字が使用されます。

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

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

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

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

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

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

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

	す。特定のデバイスに関する ioctl 呼び出しは、(そのデバイスのマニュアルページに) アルファベット順に記述されています。ioctl 呼び出しは、mtio(7I) のように、すべて io で終わる特定クラスのデバイスに使用されます。
オプション	このセクションでは、コマンドオプションの一覧と各オプションの機能のサマリーを示します。オプションは、「形式」のセクションで示されている順序で、文字通りに記述されています。オプションに引数がある場合はそのオプションの下で説明され、該当する場合はデフォルト値が示されます。
オペランド	このセクションでは、コマンドのオペランドの一覧を示し、各オペランドがコマンドの動作にどのように影響を及ぼすかを説明します。
出力	このセクションでは、コマンドによって生成される出力 (標準出力、標準エラー、または出力ファイル) について説明します。
戻り値	値を返す関数のマニュアルページの場合、このセクションには、これらの値の一覧および値が返される条件が示されます。関数が 0 や -1 などの定数値だけを返す場合は、これらの値がタグ付き段落で一覧表示されます。それ以外の場合は、各関数の戻り値が 1 つの段落で説明されます。void と宣言された関数は値を返さないため、「戻り値」では扱われません。
エラー	エラー発生時、ほとんどの関数はエラーコードを大域変数 errno に格納し、エラーの理由を示します。このセクションでは、関数が生成しうるすべてのエラーコードの一覧をアルファベット順で示し、各エラーの発生条件について説明します。複数の条件が同じエラーの原因になる場合、そのエラーコードの下に各条件が別々の段落で記述されます。
使用法	このセクションでは、詳細な説明を必要とする特殊な規則、機能、およびコマンドを示します。次に示すサブセクションは、組み込まれている機能を説明するために使用されます。 コマンド 修飾子 変数 式 入力文法
使用例	このセクションでは、コマンドまたは関数の使用例または使用方法を説明します。可能な場合は、コマンド行の入力とマシンの応答を含む完全な例が示されます。例の中で、プロンプトは example% と表示されるか、スーパーユーザーでなければならな

	い場合は example# と表示されます。例のあとに、説明、変数置換規則、または戻り値が示されます。ほとんどの例が、「形式」、「機能説明」、「オプション」、「使用法」のセクションの概念を示す実例となっています。
環境変数	このセクションでは、コマンドまたは関数が影響を与えるすべての環境変数の一覧を示し、その影響について簡単に説明します。
終了ステータス	このセクションでは、コマンドが呼び出し元のプログラムまたはシェルに返す値の一覧と、これらの値が返される条件を示します。通常、正常終了には 0 が返され、0 以外の値はそれぞれのエラー状態を示します。
ファイル	このセクションでは、マニュアルページが参照するファイル、関連ファイル、およびコマンドが作成または必要とするファイルの一覧を示します。それぞれのあとに、サマリーまたは説明があります。
属性	このセクションでは、属性タイプとその対応する値を定義することにより、コマンド、ユーティリティー、およびデバイスドライバの特性を示します。詳細については、attributes(5) を参照してください。
関連項目	このセクションでは、関連するマニュアルページ、当社のマニュアル、および外部の出版物を示します。
診断	このセクションでは、診断メッセージの一覧を示し、エラーの発生条件について簡単に説明します。
警告	このセクションでは、作業環境に重大な影響を与える可能性のある特殊な条件に関する警告の一覧を示します。これは診断の一覧ではありません。
注意事項	このセクションでは、このマニュアルページの他のどのセクションにも記載されない追加情報を示します。ユーザーへの参考情報として、特に関心を引く事項を扱います。ここで重要な情報が扱われることはありません。
使用上の留意点	このセクションでは、既知の問題について説明し、可能な場合は回避方法を示します。

参 照

紹介

名前

Intro, intro – その他の序章

機能説明

このセクションでは、次の項目について説明します。

標準　POSIX (IEEE) 標準と X/Open 仕様について `standards` で 記述します。

環境　ユーザー環境 (`environ`)、言語や文化規約によるユーザー環境のサブセット (`locale`)、大規模ファイルのコンパイル環境 (`lfcompile`)、変換のコンパイル環境 (`lfcompile64`) について記述します。

マクロ　リファレンスマニュアルのページ (`man` と `mansun`) を書式化するマクロ、およびその他のテキストを書式化するマクロ (`me`、`mm`、`ms`) について記述します。

文字　文字セット (`ascii`、`charmap`、`eqnchar`、`iconv`) のテーブル、ファイルの書式化設定 (`formats`)、ファイル名のパターンマッチング (`fnmatch`)、正規表現 (`regex` と `regexp`) について記述します。

謝辞

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

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

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

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

This notice shall appear on any product containing this material.

参 照

Standards, Environments, and Macros

名前

acl – アクセス制御リスト

機能説明

アクセス制御リスト (ACL) は、ファイルおよびディレクトリへのアクセスを付与および拒否する随意アクセス制御メカニズムです。2 つの異なる ACL モデルが Solaris リリース (POSIX ドラフト ACL および NFSv4 ACL) でサポートされます。

古い POSIX ドラフトモデルは、UFS ファイルシステムでサポートされます。このモデルは、標準化されなかった削除済みの ACL POSIX 仕様に基づいています。これは、その後 POSIX 委員会によって削除されました。

他のモデルは、NFSv4 タスクグループの標準に基づいており、Internet Engineering Task Force (IETF) から承認済みの標準です。ZFS ファイルシステムは NFSv4 モデルを使用し、POSIX ドラフトモデルより豊富なセマンティクスときめ細かなアクセス権機能を備えています。

POSIX ドラフト ACL

POSIX ドラフト ACL は、Solaris リリースで基本 UNIX ファイルアクセス権に代替のセキュリティーメカニズムを提供します。目的は、ファイルおよびディレクトリへのアクセスをさらに制限すること、または特定のユーザーにアクセス権を拡張することです。ACL を使用して、標準の所有者とグループのアクセス権、およびファイルのモードのその他のクラスビットを変更できます。ACL は、ファイルへの追加のユーザーおよびグループアクセスを付与できます。ディレクトリは、デフォルト ACL という特殊な種類の ACL を持つことができます。これは、ディレクトリの子孫によって継承される ACL エントリを定義します。POSIX ドラフト ACL には、マスクという ACL エントリがあります。マスクは、追加のユーザーとグループエントリに付与できる最大のアクセス権を定義します。chmod(1) または chmod(2) によってファイルが作成されるか、そのモードが変更されるたびに、マスクが再計算されます。これは、chmod(2) に渡されるモードで定義されたグループアクセス権になるよう再計算されます。

POSIX ドラフト ACL モデルは、従来の UNIX アクセス権の標準の rwx モデルを使用します。

ACL は次のように表されます:

acl_entry[,*acl_entry*] . . .

それぞれの *acl_entry* には ACL エントリが 1 つ含まれています。ACL エントリは、2 つまたは 3 つのコロン区切り (:) フィールドで表されます。

user:[*uid*]:*perms*　　*uid* がブランクの場合、ファイル所有者を表します。

group:[*gid*]:*perms*　　*gid* がブランクの場合、所有するグループを表します。

other:*perms*　　ファイルのその他のクラスを表します。

mask:*perms*　　付与する MAX アクセス権を定義します。

たとえば、ユーザー joe に読み取り/書き込みアクセス権を付与するには、次のように ACL エントリを指定します:

```
user:joe:rw-
```

NFSv4 ACL

NFSv4 ACL モデルは、大まかに Windows NT ACL モデルに基づいています。NFSv4 ACL は、POSIX ドラフト ACL より大幅に豊富な ACL モデルを提供します。

NFSv4 と POSIX ドラフト ACL の主な違いは次のとおりです:

- NFSv4 ACL は、rwx モデルよりきめ細かなアクセス権を提供します。
- NFSv4 ACL では、ALLOW と DENY の両方をエントリが可能です。
- NFSv4 ACL には、豊富な継承セマンティクスセットが用意されています。POSIX ACL には継承も用意されていますが、NFSv4 モデルを使用すると次の継承機能を制御できます:
  - 継承がファイルとディレクトリの両方にカスケードするか、ファイルまたはディレクトリのみにカスケードするか。
  - ディレクトリの場合、継承がディレクトリ自体に適用されるのか、1 つのレベルのサブディレクトリのみに適用されるのか、ディレクトリのすべてのサブディレクトリにカスケードされるのかを指定できます。
- NFSv4 ACL には、システムの監査トレールに接続するためのメカニズムが用意されています。現在、Solaris ではこのメカニズムはサポートされていません。
- NFSv4 ACL を使用すると、管理者は ACL エントリが検査される順序を指定できます。POSIX ドラフト ACL では、ファイルシステムは、ACL エントリを適切に定義された厳格なアクセスチェック順序に並べ替えます。

POSIX ドラフト ACL セマンティクスは、NFSv4 ACL を使用して実現できます。ただし、一部の NFSv4 ACL のみを同等の POSIX ドラフト ACL に変換できます。

アクセス権を指定する際には、異なる 3 つの chmod ACL 形式を使用できます。 冗長、コンパクト、定位置です。冗長形式では、単語を使ってアクセス権を指定し、複数のアクセス権はスラッシュ (/) 文字で区切ります。コンパクト形式ではアクセス権文字を使用し、定位置形式ではアクセス権文字またはハイフン (-) を使用してアクセス権がないことを示します。

冗長モード用のアクセス権とコンパクトモードおよび定位置モード用の短縮形について、次に説明します。短縮形は括弧内に示しています。

read_data (r)	ファイルのデータを読み取るアクセス権
list_directory (r)	ディレクトリの内容を表示するためのアクセス権。
write_data (w)	ファイルのオフセット範囲内の任意の場所にある、ファイルのデータを変更するアクセス権。これには、ファイルを拡張または任意のオフセットに書き込む能力が含まれます。

add_file (w)	ディレクトリに新しいファイルを追加するためのアクセス権。
append_data (p)	ファイルのデータを変更する能力。ただし、変更できるのは EOF 以降のみです。現在、この権限はサポートされていません。
add_subdirectory (p)	ディレクトリにサブディレクトリを作成する権限。
read_xattr (R)	ファイルの拡張属性を読み取ったり拡張属性ディレクトリの検索を実行したりする能力。
write_xattr (W)	拡張属性を作成したり拡張属性ディレクトリに書き込んだりする能力。
execute (x)	ファイルを実行する権限。
read_attributes (a)	ファイルの基本属性 (ACL 以外) を読み取る機能。基本属性は、stat レベル属性であるとみなされます。このアクセスマスクビットを許可することは、エンティティが ls(1) および stat(2) を実行できることを意味します。
write_attributes (A)	ファイルまたはディレクトリに関連付けられた時間を任意の値に変更するためのアクセス権。
delete (d)	ファイルを削除するアクセス権。
delete_child (D)	ディレクトリ内のファイルを削除する権限。
read_acl (c)	ACL を読み取るアクセス権。
write_acl (C)	ACL を書き込むアクセス権、または chmod(1) または setfacl(1) を実行する能力。
write_owner (o)	所有者を変更するアクセス権、または chown(1) または chgrp(1) を実行する能力。
synchronize (s)	同期読み取り権と書き込み権を使ってサーバー上のファイルにローカルアクセスするアクセス権。現在、この権限はサポートされていません。

次の継承フラグが NFSv4 でサポートされます:

file_inherit (f)	ディレクトリ内で新しく作成したすべてのファイルに継承します。
dir_inherit (d)	ディレクトリ内で新しく作成したすべてのディレクトリに継承します。
inherit_only (i)	ディレクトリに入れますが、そのディレクトリ自体には適用せず、新しく作成されたファイルとディレクトリのみに

	適用します。このフラグでは、継承する内容を示すために file_inherit と dir_inherit の両方またはどちらかが必要です。
no_propagate (n)	ディレクトリに入れて、ツリーの 1 レベルのみ ACL エントリを継承すべきであることを示します。このフラグでは、継承する内容を示すために file_inherit と dir_inherit の両方またはどちらかが必要です。
successful_access (S))	正常にアクセスしたときに、アラームまたは監査記録を開始するかどうかを指定します。監査/アラーム ACE タイプと同時に使用されます。
failed_access (F)	アクセスに失敗したときに、アラームまたは監査記録を開始するかどうかを指定します。監査/アラーム ACE タイプと同時に使用されます。
inherited (I)	ACE が継承されました。
-	アクセス権は付与されていません。

NFSv4 ACL は次の構文を使用して表されます:

acl_entry[,*acl_entry*]...

```
owner@:<perms>[:inheritance flags]:<allow|deny>
group@:<perms>[:inheritance flags]:<allow|deny>
everyone@:<perms>[:inheritance flags]:<allow|deny>
user:<username>[:inheritance flags]:<allow|deny>
group:<groupname>[:inheritance flags]:<allow|deny>
```

owner@	ファイル所有者
group@	グループ所有者
user	特定のユーザーのアクセス権
group	特定のグループのアクセス権

アクセス権と継承フラグは / 文字で区切られます。

ACL の指定例:

```
user:fred:read_data/write_data/read_attributes:file_inherit:allow
owner@:read_data:allow,group@:read_data:allow,user:tom:read_data:deny
```

コンパクト ACL 形式では、14 個の一意の文字を使ってアクセス権を指定します。

定位置 ACL 形式では、ls -V の形式に似た定位置引数として、アクセス権を指定します。ハイフン (-) は、その位置のアクセス権が許可されていないことを示しますが、このハイフンは省略可能であり、指定する必要があるのは必須文字だけです。

前述の文字は、定位置表記で指定する順番に並んでいます。

これらの文字を使用して、次のような方法でアクセス権を指定できます。

```
user:fred:rw------R------:file_inherit:allow
```

または、- を削除して詰めることができます。

```
user:fred:rwR:file_inherit:allow
```

継承フラグは、次のようにコンパクトな方法で指定することもできます:

```
user:fred:rwR:f:allow
user:fred:rwR:f------:allow
```

シェルレベルの Solaris API

Solaris コマンドインタフェースでは、ACL の操作がサポートされます。次の Solaris ユーティリティーが両方の ACL モデルに対応します:

`chmod` chmod ユーティリティーは、ACL の設定と削除を行うことができるように拡張されました。これは、ACL の操作をサポートする symbolic-mode 引数を拡張することで可能になります。詳細は、chmod(1) を参照してください。

`compress` ファイルを圧縮すると、元のファイルに関連付けられた ACL は、圧縮済みのファイルとともに保持されます。

`cp` デフォルトでは、-p オプションを指定しない限り、cp は ACL を無視します。-p を指定すると、所有者とグループ ID、アクセス権モード、変更時間とアクセス時間、ACL、および拡張属性 (該当する場合) が保持されます。

`cpio` -P オプションを指定すると、ACL が保持されます。

`find` -acl フラグを指定すると、ACL が設定されたファイルが検索されます。

`ls` デフォルトでは、ls は ACL 情報を表示しません。-v オプションを指定すると、ファイルの ACL が表示されます。

`mv` ファイルを移動すると、すべての属性が名前変更されたファイルとともに移動されます。ファイルシステムの境界の間でファイルを移動すると、ACL が複製されます。ACL 情報を複製できない場合、移動は失敗し、ソースファイルは削除されません。

`pack` ファイルをパックすると、元のファイルに関連付けられた ACL は、パック済みのファイルとともに保持されます。

`rcp` rcp は、コピーをサポートするために拡張されました。リモートホストで ACL がサポートされるときのみ、ファイルの ACL が保持されます。

tar　　-p オプションを指定すると、ACL が保持されます。

unpack　　ACL のあるファイルをアンパックすると、アンパック済みのファイルでは ACL 情報が保持されます。

アプリケーションレベルの API

プログラムレベルでファイルシステム ACL にアクセスするために必要なプライマリインタフェースは、acl_get() 関数と acl_set() 関数です。これらの関数は、POSIX ドラフト ACL と NFSv4 ACL の両方をサポートします。

ファイルの ACL を取得する

```
int acl_get(const char *path, int flag, acl_t **aclp);
int facl_get(int fd, int flag, acl_t **aclp);
```

acl_get(3SEC) および facl_get(3SEC) 関数は、パスによって名前が付けられるか、開いているファイル記述子 fd によって名前が参照されるファイルで ACL を取得します。フラグ引数は、簡易 ACL を取得すべきかどうかを指定します。フラグ引数が ACL_NO_TRIVIAL の場合、簡易ではない ACL のみが取得されます。ACL は aclp 引数で返されます。

ACL 構造を解放する

```
void acl_free(acl_t *aclp)s;
```

acl_free() 関数は、引数 aclp; で割り当てられたメモリを解放します。

ファイルで ACL を設定する

```
int acl_set(const char *path, acl_t *aclp);
int facl_set(int fd, acl_t *aclp);
```

パスによって名前が付けられるか、開いているファイル記述子 fd によって名前が参照されるファイルで ACL を設定するには、acl_set(3SEC) および facl_get(3SEC) 関数が使用されます。aclp 引数は、設定する ACL を指定します。ターゲットファイルシステムで NFSv4 ACL がサポートされるとき、acl_set(3SEC) は、POSIX ドラフト ACL を NFSv4 ACL に変換します。POSIX ドラフト ACL でサポートされるファイルシステムで NFSv4 ACL を設定しようとすると、変換は実行されません。

ACL の簡易性を判定する

```
int acl_trivial(const char *path);
```

acl_trivial() 関数は、ファイルに簡易 ACL があるかどうかを判定するために使用されます。ACL が簡易かどうかは、ACL のタイプに基づきます。POSIX ドラフト ACL は、MIN_ACL_ENTRIES より大きい場合は簡易ではありません。NFSv4/ZFS スタイル ACL は、owner@、group@、および everyone@ より大きいエントリを持つか、継承フラグが設定されているか、POSIX アクセス制御の要件を満たす方法で順序付けられていない場合は簡易ではありません。

ファイルから ACL をすべて削除する

```
int acl_strip(const char *path, uid_t uid, gid_t gid, mode_t mode);
```

acl_strip() 関数は、ファイルからすべての ACL を削除し、引数モードで渡される内容に基づいて簡易 ACL と置き換えます。ACL の置き換え後に、ファイルの所有者とグループは、uid および gid パラメータで指定された値に設定されます。

外部表現間で ACL を変換する

```
int acl_fromtext(const char *path, acl_t **aclp);
char *acl_totext(acl_t *aclp, int flags);
```

`acl_text()` 関数は、aclp が指し示す内部 ACL 表現を外部表現に変換します。外部表現の詳細は、「説明」を参照してください。

`acl_fromtext()` 関数は、外部表現を内部表現に変換します。外部表現の詳細は、「説明」を参照してください。

使用例

次の例は、API を使用して ACL で基本操作を実行する方法を示しています。

例 1 ACL を取得および設定する

ACL を取得して、別のファイルで設定するには、次を使用します:

```
error = acl_get("file", ACL_NO_TRIVIAL, &aclp);

if (error == 0 && aclp != NULL) {
error = acl_set("file2", aclp)
acl_free(aclp);
}
...
```

例 2 任意の ACL を取得および設定する

簡易 ACL を含む任意の ACL を取得して、別のファイルで設定するには、次を使用します:

```
error = acl_get("file3", 0, &aclp);
if (error == 0) {
error = acl_set("file4", aclp)
acl_free(aclp);
}
...
```

例 3 ファイルに簡易 ACL があるかどうかを判定する

ファイルに簡易 ACL があるかどうかを判定するには、次を使用します:

```
istrivial = acl_trivial("file")

if (istrivial == 0)
printf("file %s has a trivial ACL\n", file);
else
printf("file %s has a NON-trivial ACL\n", file);
...
```

例 4　ファイルから ACL をすべて削除する

ファイルからすべての ACL を削除して、新しいモード、所有者、およびグループを設定するには、次を使用します:

```
error = acl_strip("file", 10, 100, 0644);
...
```

関連項目

chgrp(1), chmod(1), chown(1), cp(1), cpio(1), find(1), ls(1), mv(1), tar(1), setfacl(1), chmod(2), acl(2),stat(2),acl_get(3SEC), aclsort(3SEC), acl_fromtext(3SEC), acl_free(3SEC), acl_strip(3SEC), acl_trivial(3SEC)

名前

ad – ネーミングリポジトリとしての Active Directory

機能説明

Solaris クライアントは、Active Directory (AD) サーバーからネーミング情報を取得できます。

最初に Solaris システムを AD ドメインに参加させ、次に `ad` キーワードを `nsswitch.conf(4)` ファイルの適切なエントリに追加する必要があります。Solaris システムを AD ドメインに参加させるには、`kclient(1M)` ユーティリティーを使用します。AD ネームサービスは、`passwd` および `group` に対するネーミングデータベースをサポートしています。

Windows ユーザーはログインできません。`user_attr(4)` データベースには Windows ユーザーのエントリがなく、`passwd(1)` コマンドは、AD とのユーザーパスワードの同期をサポートしていません。

Solaris AD クライアントは、自動検出テクノロジを使用して、ドメインコントローラやグローバルカタログサーバーなどの AD ディレクトリサーバーを見つけます。クライアントはまた、LDAP v3 プロトコルを使用して、AD サーバーのネーミング情報にアクセスします。AD サーバー方式は、AD クライアントがネイティブな AD 方式で動作しているので、変更を必要としません。Solaris AD クライアントは、`idmap(1M)` サービスを使用して、Windows セキュリティー識別子 (SID) と Solaris ユーザー識別子 (UID) およびグループ識別子 (GID) とを対応付けます。ユーザー名とグループ名は、AD ユーザーおよびグループオブジェクトの `sAMAccountName` 属性から取得され、オブジェクトが存在するドメインでタグ付けされます。ドメイン名は、`@` 文字でユーザー名またはグループ名から切り離されます。

クライアントは SASL/GSSAPI/KRB5 セキュリティーモデルを使用します。クライアントを AD に参加させるには、`kclient` ユーティリティーが使用されます。参加操作中、`kclient` はクライアント上で Kerberos v5 を構成します。`kclient(1M)` を参照してください。

ファイル

`/etc/nsswitch.conf`	ネームサービススイッチの構成ファイル。
`/etc/nsswitch.ad`	ad、dns、および files で構成されたネームサービススイッチのサンプル構成ファイル。
`/usr/lib/nss_ad.so.1`	AD 用のネームサービススイッチモジュール。

関連項目

`passwd(1)`, `svcs(1)`, `idmap(1M)`, `idmapd(1M)`, `kclient(1M)`, `svcadm(1M)`, `svccfg(1M)`, `svccfg(1M)`, `nsswitch.conf(4)`, `user_attr(4)`, `smf(5)`

名前

ars – receive Solaris audit logs from a remote server

形式

`/usr/sbin/auditd`

機能説明

Audit Remote Server (ARS) is the counterpart of the `audit_remote(5)` plugin. Data sent by the plugin can be captured, processed, and stored by the server according to the its configuration.

ARS is delivered as a disabled Solaris audit component. It is necessary to configure ARS before it can be used to process a remote audit trail. ARS configuration is twofold:

- the underlying security mechanisms used for secure audit data transport has to be configured (see `audit_remote(5)`);
- the audit remote subsystem has to be configured.

To observe and configure the ARS, use the `auditconfig(1M)` `-setremote` and `-getremote` options. The configuration is divided between the configuration of *server* and *group*. The *server* configuration allows for changing common ARS parameters, while the *group* keyword allows configuration of connection groups, the sets of hosts sharing the same local storage parameters.

Server configuration attributes

listen_address
: The address the server listens on. An empty *listen_address* attribute defaults to listen on all local addresses.

listen_port
: The local listening port; 0 defaults to 16162, the port associated with the “solaris-audit” Internet service name. See `services(4)`.

login_grace_time
: The server disconnects after login grace time (in seconds) if the connection has not been successfully established; 0 defaults to no limit.

max_startups
: The number of concurrent unauthenticated connections to the server at which the server starts refusing new connections. The value might be specified in *begin*:*rate*:*full* format to allow random early drop mode, for example 10:30:60, meaning that ARS would refuse connection attempts with a probability of *rate*/100 (30% in our example) if there are currently 10 (from the *start* field) unauthenticated connections. The probability increases linearly and all connection attempts are refused if the number of unauthenticated connections reaches *full* (60 in our example).

Group configuration attributes

The *binfile_dir*, *binfile_fsize*, and *binfile_minfree* attributes follow the respective *p_** attributes defined in `audit_binfile(5)`. Brief descriptions follow.

binfile_dir
: The directory for storing per host audit data.

binfile_fsize
The maximum size of each of the stored audit trail files; 0 defaults to no limit.

binfile_minfree
The minimum free space on file system with *binfile_dir* before the `audit_binfile` informs the administrator via `audit_warn(1M)`; 0 defaults to no limit.

hosts
The hosts in the given connection group allowed to send audit data to server. A comma is a delimiter in case of multiple host entries. If *hosts* is empty, such connection group is called a wild card connection group. If a new connection cannot be classified to any other (non-wild card) connection group and there is an active wild card connection group configured, the new connection is classified to that connection group. Only one active wild card connection group can be configured.

For comprehensive configuration description and examples, see the section on Auditing in 『Oracle Solaris 11.1 の管理: セキュリティーサービス』.

使用例

例 1 Audit Remote Server configuration

The following example describes steps to configure audit remote server to listen on specific address. One wild card and one non-wild card connection group will be created. The non-wild card connection group configuration will address remote audit data from `tic.cz.example.com` and `tac.us.example.com`, the trail will be stored in `/var/audit/remote`.

```
# Print the current audit remote server configuration.
# Both server and connection groups (if any) is displayed.
auditconfig -getremote

# Set address the audit remote server will listen on.
auditconfig -setremote server "listen_address=192.168.0.1"

# Create two connection groups. Note that by default the
# connection group is created with no hosts specified
# (wild card connection group).
auditconfig -setremote group create clockhouse
auditconfig -setremote group create sink

# Add hosts to the connection group (convert the wild card
# connection group no non-wild card one). Set the storage
# directory and activate the connection group.
auditconfig -setremote group active clockhouse \
    "hosts=tic.cz.example.com,tac.us.example.com,
    binfile_dir=/var/audit/remote"

# Activate the wild card connection group.
auditconfig -setremote group active sink
```

例 1 Audit Remote Server configuration *(続き)*

```
# Verify the audit remote server configuration.
auditconfig -getremote

# Start or refresh the audit service.
audit -s
```

属性

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

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

関連項目

audit(1M), auditconfig(1M), auditd(1M), audit_warn(1M), services(4), attributes(5), audit_binfile(5), smf(5)

See the section on Auditing in 『Oracle Solaris 11.1 の管理: セキュリティーサービス』.

注意事項

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

名前

ascii – map of ASCII character set

形式

```
cat /usr/pub/ascii
```

機能説明

`/usr/pub/ascii` is a map of the ASCII character set, to be printed as needed. It contains octal and hexadecimal values for each character. While not included in that file, a chart of decimal values is also shown here.

Octal — Character

```
000 NUL  001 SOH  002 STX  003 ETX  004 EOT  005 ENQ  006 ACK  007 BEL
010 BS   011 HT   012 NL   013 VT   014 NP   015 CR   016 SO   017 SI
020 DLE  021 DC1  022 DC2  023 DC3  024 DC4  025 NAK  026 SYN  027 ETB
030 CAN  031 EM   032 SUB  033 ESC  034 FS   035 GS   036 RS   037 US
040 SP   041 !    042 "    043 #    044 $    045 %    046 &    047 ’
050 (    051 )    052 *    053 +    054 ,    055 −    056 .    057 /
060 0    061 1    062 2    063 3    064 4    065 5    066 6    067 7
070 8    071 9    072 :    073 ;    074 <    075 =    076 >    077 ?
100 @    101 A    102 B    103 C    104 D    105 E    106 F    107 G
110 H    111 I    112 J    113 K    114 L    115 M    116 N    117 O
120 P    121 Q    122 R    123 S    124 T    125 U    126 V    127 W
130 X    131 Y    132 Z    133 [    134 \    135 ]    136 ^    137 _
140 ‘    141 a    142 b    143 c    144 d    145 e    146 f    147 g
150 h    151 i    152 j    153 k    154 l    155 m    156 n    157 o
160 p    161 q    162 r    163 s    164 t    165 u    166 v    167 w
170 x    171 y    172 z    173 {    174 |    175 }    176 ~    177 DEL
```

Hexadecimal — Character

```
00 NUL   01 SOH   02 STX   03 ETX   04 EOT   05 ENQ   06 ACK  07 BEL
08 BS    09 HT    0A NL    0B VT    0C NP    0D CR    0E SO   0F SI
10 DLE   11 DC1   12 DC2   13 DC3   14 DC4   15 NAK   16 SYN  17 ETB
18 CAN   19 EM    1A SUB   1B ESC   1C FS    1D GS    1E RS   1F US
20 SP    21 !     22 "     23 #     24 $     25 %     26 &    27 ’
28 (     29 )     2A *     2B +     2C ,     2D −     2E .    2F /
30 0     31 1     32 2     33 3     34 4     35 5     36 6    37 7
38 8     39 9     3A :     3B ;     3C <     3D =     3E >    3F ?
40 @     41 A     42 B     43 C     44 D     45 E     46 F    47 G
48 H     49 I     4A J     4B K     4C L     4D M     4E N    4F O
50 P     51 Q     52 R     53 S     54 T     55 U     56 V    57 W
58 X     59 Y     5A Z     5B [     5C \     5D ]     5E ^    5F _
60 ‘     61 a     62 b     63 c     64 d     65 e     66 f    67 g
68 h     69 i     6A j     6B k     6C l     6D m     6E n    6F o
70 p     71 q     72 r     73 s     74 t     75 u     76 v    77 w
78 x     79 y     7A z     7B {     7C |     7D }     7E ~    7F DEL
```

Decimal — Character

```
  0 NUL    1 SOH    2 STX    3 ETX    4 EOT    5 ENQ    6 ACK    7 BEL
  8 BS     9 HT    10 NL    11 VT    12 NP    13 CR    14 SO    15 SI
```

```
 16 DLE  17 DC1   18 DC2   19 DC3   20 DC4   21 NAK   22 SYN   23 ETB
 24 CAN  25 EM    26 SUB   27 ESC   28 FS    29 GS    30 RS    31 US
 32 SP   33 !     34 "     35 #     36 $     37 %     38 &     39 ’
 40 (    41 )     42 *     43 +     44 ,     45 −     46 .     47 /
 48 0    49 1     50 2     51 3     52 4     53 5     54 6     55 7
 56 8    57 9     58 :     59 ;     60 <     61 =     62 >     63 ?
 64 @    65 A     66 B     67 C     68 D     69 E     70 F     71 G
 72 H    73 I     74 J     75 K     76 L     77 M     78 N     79 O
 80 P    81 Q     82 R     83 S     84 T     85 U     86 V     87 W
 88 X    89 Y     90 Z     91 [     92 \     93 ]     94 ^     95 _
 96 ‘    97 a     98 b     99 c    100 d    101 e    102 f    103 g
104 h   105 i    106 j    107 k    108 l    109 m    110 n    111 o
112 p   113 q    114 r    115 s    116 t    117 u    118 v    119 w
120 x   121 y    122 z    123 {    124 |    125 }    126 ~    127 DEL
```

ファイル

`/usr/pub/ascii` On-line chart of octal and hexadecimal values for the ASCII character set.

名前

attributes, architecture, availability, CSI, stability, MT-Level, standard – インタフェースの属性

機能説明

マニュアルページの「属性」セクションには、属性タイプと対応する値を定義する表が含まれます。属性表の例を次に示します。すべての属性タイプがすべての種類のインタフェースに適合するわけではありません。

属性タイプ	属性値
アーキテクチャー	SPARC
使用条件	system/kernel
CSI	有効
インタフェースの安定性	確実
MT レベル	安全
標準	standards(5) を参照してください。

アーキテクチャー

アーキテクチャーはプロセッサまたは固有のハードウェアを定義します。uname(1) の -p オプションを参照してください。必要なアダプタまたは周辺機器を示す場合もあります。

使用条件

これは、マニュアルページで説明されているコマンドまたはコンポーネントを含むソフトウェアパッケージを示します。このコマンドを使用できるようにするには、示されたパッケージがインストールされている必要があります。パッケージの追加方法についての詳細は、pkg(1) を参照してください。

コードセットの独立性 (CSI)

どのコードセットの特性にも依存していない OS ユーティリティーおよびライブラリは、コードセットの独立性 (CSI) を保持している、と言われます。これらは、CSI が有効な属性を持っています。これは、たとえば拡張 UNIX コードセット (EUC) でのみ動作するような、多くのコマンドやユーティリティーとは対照的です。EUC は、同時に 4 つまでのコードセットをサポートできるエンコーディング方式で、一般にアジア地域の文字セットを表すために使用されます。

ただし、実用上の理由によりこの独立性は絶対ではありません。現在の CSI 実装には、まだいくつかの仮定が適用されています。

- ファイルコードは ASCII のスーパーセットです。
- 複数バイト文字および NULL で終わる UNIX ファイル名をサポートするため、NULL および / (スラッシュ) 文字を、複数バイト文字の一部に含めることはできません。
- 「ステートレス」ファイルコードのエンコーディングだけがサポートされています。ステートレスなエンコーディングでは、単一のシフトは除外されていませんが、シフト、ロッキングシフト、指示、呼び出しなどは避けられています。

- プロセスコード (wchar_t 値) は実装に依存し、時期によって、または実装間やロケール間で、異なることがあります。
- すべてのオブジェクトが、任意の文字で構成される名前を持つことができるわけではありません。次のオブジェクトは、ASCII 文字から構成される名前にする必要があります。
  - ユーザー名、グループ名、およびパスワード
  - システム名
  - プリンタおよび特殊デバイスの名前
  - 端末の名前 (/dev/tty*)
  - プロセス ID 番号
  - メッセージキュー、セマフォー、および共有メモリーラベル。
  - 以降は、ISO Latin-1 文字または EUC 文字で構成する必要があります。
    - ファイル名
    - ディレクトリ名
    - コマンド名
    - シェルの変数および環境変数名
    - ファイルシステムへのマウントポイント
    - NIS キー名およびドメイン名
- NFS 共有ファイルの名前は、ASCII 文字で構成するようにしてください。ファイルやディレクトリの名前や内容に、ASCII 以外のコードセットの文字を使用することもできますが、ASCII コードセットのみを使用すると、ローカリゼーションに関係なくどのマシンでも NFS マウントを使用できます。CSI が有効になっているコマンドおよびユーティリティーはいずれも、2.6 でリリースされたシングルバイトおよび複数バイトのロケールを処理できます。アプリケーションが国際化サービスの完全なサポートを受けるには、動的結合を適用する必要があります。静的にリンクされたプログラムでは、C ロケールおよび POSIX ロケールだけがサポートされます。

インタフェースの安定性

Oracle Solaris は多くの場合、開発者が新しい技術をできるだけ早く評価できるようにするため、初期の段階で開発者がそれらの技術へアクセスできるようにします。残念ながら、新しい技術は変更が生じやすいため、標準化により以前のバージョンとのインタフェースの非互換性が生じることもよくあります。

リスクに対する予測を合理的に行うために、開発者は将来のリリースでインタフェースがどの程度変更される可能性があるかを知っておく必要があります。開発者がこうした予測をしやすくするため、インタフェースの安定性についての情報が、コマンド、エントリポイント、およびファイル形式に関するいくつかのマニュアルページに含まれています。

比較的安定したインタフェースについて、Oracle Solaris は、将来のマイナーリリースでも引き続き確実に動作するよう努力するため、ほぼすべてのアプリケーションで安全に使用できます。「確実」インタフェースにのみ依存するアプリ

ケーションは、確実に将来のマイナーリリースでも正常に機能し続けます (ただし過去のメジャーリリースで機能するとは限らない)。

比較的安定していないインタフェースについては、実験と試作が可能ですが、将来のマイナーリリースで互換性のない変更がなされたり、削除されたり、代替のインタフェースと置き換えられたりする可能性を理解したうえで使用するようにしてください。

Oracle Solaris がドキュメント化しない「インタフェース」(たとえば、ほとんどのカーネルデータ構造やシステムヘッダーファイルの一部のシンボル) は、実装によるアーティファクトである場合があります。そのような内部インタフェースは、互換性のない変更がなされたり削除されたりする場合があり、また、通常、リリースノートでそのような変更について言及されることはありません。

リリースレベル

互換性について検討する助けとして、製品には名前とともにリリースレベルが付与されています。各リリースレベルには、低いレベルに適合する変更も含まれます。

解放	バージョン	重要度
メジャー	x.0	重要な機能が追加され、異なる (場合によっては非互換の) 標準リビジョンに準拠していると考えられます。低い確率で、「確実」インタフェースが変更、削除、または置換される場合もあります。製品の初期リリースは通常 1.0 です。
マイナー	x.y	x.0 または以前のリリース (y!=0) と比較すると、次を含む場合があります。機能の追加、「確実」インタフェースに対する互換性のある変更、「不確実」インタフェースまたは「流動的」インタフェースに対する非互換である可能性の高い変更。
マイクロ	x.y.z	以前のリリース (z!=0) とのインタフェース互換性確保が意図されますが、バグの修正、パフォーマンスの改善、および追加ハードウェアのサポートが追加される場合もあります。「流動的」インタフェースに非互換の変更が加えられる可能性もあります。

インタフェースの安定性に関しては、更新リリース (パッチリリースと呼ばれることもある) がマイクロリリースに相当すると見なされます。

分類

次の表は、安定性レベルの分類とリリースレベルとの関係をまとめたものです。最初の列は安定性レベルです。2 番目の列は互換性のない変更のリリースレベルを、3 番目の列はその他のコメントを表します。個別の分類に関する完全な考察については、以降の該当するサブセクションを参照してください。

安定性	解放	Comments
確実	メジャー (x.0)	非互換性は例外的です。
不確実	マイナー (x.y)	非互換は一般的です。
流動的	マイクロ (x.y.z)	非互換は一般的です。

このマニュアルページで説明されているインタフェースの安定性レベルの分類は、特に明記しない限り、ソースインタフェースとバイナリインタフェースの両方に適用されます。すべての安定性レベルの分類は公開されていますが、非公開 (`Private`) 分類だけは例外です。公開インタフェース (マニュアルページにドキュメント化されたもの) の正確な安定性レベルは、明示されていない限り特定されていません。ドキュメント化されていないインタフェースの安定性レベルは、暗黙的に非公開です。

Oracle Solaris 製品のコンポーネントであるドキュメント以外のドキュメントが存在しても、Oracle Solaris 製品により提供されるインタフェースについて、いずれかの安定性レベルが暗示されているとみなすべきではありません。安定性レベルに関する唯一の情報源は、リファレンスマニュアルページです。

確実
: 「確実」インタフェースの目的は、サードパーティーがこれらのインタフェースに基づいてアプリケーションを開発してリリースできるようにし、そのインタフェースが導入された製品のリリース以降、同じメジャーリリース内であればどのリリースでも、アプリケーションの動作を保証することです。メジャーリリースでも、互換性のない変更はまれにありますが、正当な理由を必要とします。

  業界標準として定義され管理されているインタフェースは、多くの場合、「確実」インタフェースとして扱われます。この場合、管理している組織やバージョン管理された公開ドキュメントは、通常、ドキュメントの属性表やほかの部分で「標準」エントリに記述されています。

  ごくまれなことですが、互換性のない変更が加えられる可能性もあります。このような変更は、このドキュメントの「例外」セクションで説明されているように重大な欠陥がある場合は任意のリリースで、また、サポート終了プロセスに従ったマイナーリリースで発生します。「確実」インタフェースのサポートを打ち切る必要がある場合、Oracle Solaris は通知を行うよう努め、安定性レベルが「廃止」としてマークされます。

不確実
: このようなインタフェースについては、あるマイナーリリースから次のマイナーリリースへの移行で、ソースやバイナリの互換性は保証されません。インタフェースの削除といった互換性のない大幅な変更がマイナーリリースで加えられる可能性もあります。「不確実」インタフェースは、通常、リリースに依存しない製品での使用には適していません。

互換性のない変更をインタフェースに加える場合には、使いやすさの考慮なども含めて実際にインタフェースを向上させるという目的が必要です。一般的な想定として、「不確実」インタフェースに互換性のない変更が加えられる可能性は低く、そのような変更が発生してもその影響は小さく、多くの場合は軽減計画が存在します。

通常、「不確実」インタフェースは、次のいずれかのサブカテゴリに分類されます。

1. 実験的または暫定的なインタフェース。通常、外部の開発者が新しいまたは急速に変化している技術を早く利用できるようにするため、あるいは、より一般的なソリューションが将来予期できる場合に、問題に対する暫定的なソリューションを提供するために使用されます。
2. 外部組織によって仕様が管理されているインタフェース。Oracle Solaris は、外部仕様と同期をとれる次のマイナーリリースまで、以前のリリースとの互換性を保つために適切な努力をします。
3. 安定性より革新性 (および使いやすさ) を重視するユーザーを対象としたインタフェース。この属性は、より高い層のコンポーネントの管理インタフェースに関連付けられることがよくあります。

「不確実」インタフェースの場合、Oracle Solaris は 2 つのマイナーリリース間でのソースまたはバイナリのどちらの互換性に関しても保証しません。これらのインタフェースに基づいて開発されるアプリケーションは、将来のマイナーリリースで動作しなくなる場合があります。

流動的
「流動的」インタフェースは、任意の理由でいつでも変更される可能性があります。

Oracle Solaris 製品では、「流動的」というインタフェースの安定性レベルを使用することで、急速に変化する流動的な仕様にすばやく追従できます。多くの場合、この方が利用者の期待に適切に応えることになるため、インタフェースの安定性の向上よりも優先されます。

通常、この分類レベルは、Oracle Solaris 以外の組織によって管理されているインタフェースに適用されます。ただし、標準化団体やフリーオープンソースソフトウェア (FOSS) コミュニティーによって管理されている、インタフェースの互換性を重視する仕様とは異なり、互換性のない変更がインタフェース仕様に加えられることは非常にまれであるとは断言できません。また、ユーザーに安定性を提供することよりも、最小限の遅延でコミュニティーに追従することの方が重要視される場合には、FOSS によって管理されているソフトウェアに適用されることもあります。

さらに、「流動的」という分類レベルは、信頼できる組織や広く認められている組織による定義の過程にあるインタフェースに適用されることもよくあります。これらはドラフト標準と総称されます。「IETF インターネットドラフト」は、よく理解されている開発中の仕様の例です。

「流動的」は、実験的なインタフェースにも適用できます。

「流動的」インタフェースの場合、パッチも含む 2 つのリリース間でのソースやバイナリの互換性は保証されません。このようなインタフェースを使用しているアプリケーションは、将来のリリースでは正しく機能しない可能性があります。

インタフェースではない
この状況は、インタフェースであると推測できるが実際にはインタフェースでないエンティティーが存在する場合に発生することがあります。一般的な例として、人による使用だけを目的とした CLI 出力や GUI の正確なレイアウトなどがあります。

このような混乱が発生しやすいと特定される状況では、この分類を使用して簡単に明確にすることができます。エンティティーにこの分類が適用されていないことは、そのエンティティーが何らかの形式のインタフェースであることを示すものではありません。混乱の可能性が特定されていないことだけを示しています。

非公開
非公開インタフェースは、提供元のコンポーネント (または製品) 自体による使用のみが意図されたインタフェースです。それでも非公開インタフェースは、ほかのコンポーネントから見えたりアクセスできたりする場合があります。ほかのコンポーネントの非公開インタフェースを使用することは安定性の大きなリスクが伴うため、そのような使用は明示的にサポートされていません。Oracle Solaris によって提供されていないコンポーネントは、非公開インタフェースを使用するべきではありません。

ほとんどの非公開インタフェースはドキュメント化されていません。非公開インタフェースがドキュメント化されているのは例外的です。非公開インタフェースがドキュメント化される理由として、公開安定性レベルのいずれかに再分類される予定がある、普及度が非常に高い、などが挙げられます。

廃止 (Obsolete)
「廃止」という修飾子は、前述の分類レベルとともに使用できます。「廃止」修飾子は、「非推奨」になった、または一般的な使用には推奨されなくなったインタフェースを示します。既存のインタフェースは、「廃止」修飾子の適用により、ほかのステータス (「確実」や「不確実」など) から降格されることがあります。これにより、そのインタフェースが削除される (または互換性のない変更が加えられる) 前に、そのインタフェースからの移行をユーザーに促します。

廃止インタフェースは現在のリリースではサポートされていますが、将来の (マイナー) リリースでは削除される予定です。インタフェースのサポートが中止されるとき、Oracle Solaris はサポートを中止する前に通知を行うよう努めます。廃止インタフェースを使用すると、警告メッセージが表示される場合があります。

例外

まれに、インタフェースの安定性に関する保証を破棄することが Oracle Solaris とユーザーの両者にとって最大の利益になる場合があります。インタフェースの提供者がインタフェースの安定性に関する保証に違反する一般的な理由を、次の表にいくつか示します。

1. インタフェースに脆弱性が内在することによるセキュリティーホール。
2. インタフェースに脆弱性が内在することによるデータ破壊。
3. 適合性テストの解釈の変更や強化によって明らかになる標準違反。
4. Oracle Solaris で管理されていないインタフェース仕様に互換性のない変更が加えられ、インタフェース利用者の大多数が新しいインタフェースを求めている。
5. 互換性のない変更を行わないとユーザーの理解が得られない。たとえば、DOS 8.3 の命名規則の制限が廃止されたとき、pcfs に互換性のない変更が加えられました。

互換性のない変更が例外として認められる場合、変更は常に「できるだけ主要な」リリース手段で配布されます。ただし、脆弱性やブランド契約の要件のため、やむを得ずパッチで配布されることもあります。

以前のインタフェース分類方式との互換性

Solaris 10 およびそれより前のリリースでは、異なるインタフェース分類方式が使用されていました。以前の分類方式と新しい分類方式のマッピングを次の表にまとめます。

旧	新	Comments
標準	確実	属性表の「標準」属性タイプのエントリが表示されます。
安定	確実	名前が変更されました。
開発中	不確実	実際の保証が一致します。
変更の可能性あり	不確実	名前が変更されました。
外部	流動的	名前が変更され、可能な使用法が拡張されました。
廃止または互換性がなくなる可能性あり	(廃止または互換性がなくなる可能性あり)	以前は分類でしたが、現在は修飾子です。

フリーオープンソースソフトウェアの重要性が高まっているため、「安定」/「不安定」から「確実」/「不確実」という名前に変更されました。「安定」という用語は、FOSS コミュニティーでの一般的な使用法と競合していました。

「開発中」の定義があいまいなため、解釈が難しくなっていました。新しい分類方式への移行に伴い、以前の「開発中」インタフェースの多くが「確実」に昇格されました。ただし、「開発中」という用語が出現した場合は、「不確実」と見なすようにしてください。

MT レベル

ライブラリは、複数のスレッドをサポートする能力に応じてカテゴリに分類されます。複数または異なるレベルの関数を含むマニュアルページでは、「注意事項」または「使用法」のセクションでこの点が説明されています。

安全
: 安全はマルチスレッドのアプリケーションから呼び出し可能なコードの属性です。安全インタフェースまたは安全コードセグメントへの呼び出しにより、複数のスレッドによって呼び出された場合でも結果が有効になります。よく見過ごされる点ですが、この安全インタフェースまたは安全コードセグメントの結果は、すべてのスレッドにグローバルな影響を及ぼす可能性があります。たとえば、あるスレッドのファイルを開いたり閉じたりするアクションは、プロセス内の他のすべてのスレッドから表示可能です。マルチスレッドのアプリケーションには、これらのインタフェースを安全な方法で使用する責任があり、これはインタフェースが安全かどうかとは異なります。たとえば、アプリケーション内の他のスレッドによってまだ使用中のファイルを閉じるマルチスレッドのアプリケーションは、close(2) インタフェースを安全に使用していません。

安全ではない
: 安全ではないライブラリには、保護されていないグローバルおよび静的なデータが含まれています。ライブラリ内で一度に 1 つのスレッドだけが実行されるようアプリケーションで取り決めない限り、このライブラリの使用は安全ではありません。安全ではないライブラリには安全な関数が含まれている場合がありますが、ほとんどのライブラリの関数は呼び出すのが安全ではありません。安全ではない一部の関数は、MT-安全である再入可能な対象を持ちます。再入可能な関数には、関数名に _r のサフィックスが付いています。

MT-安全
: MT-安全ライブラリは、マルチスレッドのアクセスに対する準備が整っています。ロックによってグローバルおよび静的なデータを保護し、適度な量の並行性を実現しています。安全に使用できるライブラリも、MT-安全であるとは限りません。たとえば、ライブラリ全体をモニターで囲むとライブラリは安全になりますが、並行性をサポートしないため MT-安全とはみなされません。MT-安全ライブラリは適度な量の並行性を許容する必要があります。(この定義の目的は、ライブラリが安全であるとされる際に、その意味するものを明確にすることです。安全なライブラリの定義では、ライブラリが並行性をサポートするかどうかは示しません。MT-安全の定義では、ライブラリが安全で、ある程度の並行性をサポートすることを明確にしています。つまり安全の定義では、シングルスレッドを意味する場合も、任意の程度のマルチスレッドを意味する場合もあります。)

非同期シグナル安全
: 非同期シグナル安全とは、シグナルハンドラから安全に呼び出すことのできる特定のライブラリ関数のことです。非同期シグナル安全関数を実行するスレッドは、シグナルによって割り込まれた場合に、自分自身でデッドロックすることはありません。シグナルは、ロックを取得する MT-安全関数にとってのみ問題になります。

  非同期シグナル安全関数は、MT-安全でもあります。非同期シグナル安全関数でロックが取得されると、シグナルは無効になります。これらのシグナルは、同じロックを取得する可能性のあるシグナルハンドラが呼び出されないようにします。

MT-安全の例外
: 例外の説明については、このページの「注意事項」または「使用法」のセクションを参照してください。

安全の例外
: 例外の説明については、このページの「注意事項」または「使用法」のセクションを参照してください。

Fork-安全
: fork(2) 関数は、その関数を呼び出したスレッドだけを子プロセスに複製します。fork1(2) 関数は、以前の関数との互換性のために存在しており、 `fork()` と同義です。`fork()` が呼び出されたとき、fork を実行しているスレッド以外のスレッドがロックを保持していた場合、ロックは子プロセスに引き続き保持されますが、所有者であるスレッドは複製されないため、ロックには所有者がいないことになります。ロックの取得を試みる関数を子が呼び出すと、デッドロックが発生します。

  `fork()` が呼び出されるとき、Fork-安全ライブラリはその内部ロックすべてが fork を実行するスレッドによってのみ保持されるようにします。これは通常、ライブラリの初期化時に呼び出される `pthread_atfork(3C)` により実現されます。

  `forkall(2)` 関数は、まれなケースとして、fork を実行するときにプロセスがそのすべてのスレッドを複製する必要がある場合にその機能を提供します。`forkall()` が呼び出されたときに、`pthread_atfork()` アクションは実行されません。`forkall()` を呼び出すことに関連する危険が存在します。プロセス内のスレッドで入出力操作を実行中に、別のスレッドが `forkall()` を呼び出すと、同じ入出力操作が親プロセスと子プロセスの両方で行われ続け、結果としてデータが破壊される場合があります。このような競合状態に関する理由から、`forkall()` の使用は推奨されていません。

  Solaris 10 よりも前のすべての Solaris リリースでは、`fork()` の動作はアプリケーションが `-lpthread` (POSIX スレッド、`standards(5)` を参照) とリンクしているかどうかに依存していました。`-lpthread` とリンクされている場合、`fork()` は `fork1()` と同じように動作し、リンクしていない場合は `forkall()` と同じように

動作していました。fork() の動作に関する混乱を避けるため、アプリケーションは状況に応じてfork1() またはforkall() を指定できるようになりました。

取り消し安全性
マルチスレッドアプリケーションがpthread_cancel(3C) を使用してスレッドを取り消し (すなわち終了) する場合、対象となるスレッドがロックまたは割り当てられたメモリーなどのリソースを保持したまま終了する場合があります。スレッドにリソースを適切に解放する適切な取り消しクリーンアップハンドラがインストールされていない場合 (pthread_cancel(3C) を参照)、アプリケーションは「取り消し非安全」つまり取り消しに関して安全ではないことになります。このように安全でないと、取り消されたスレッドのロックが解放されないため、またはリソースリーク (たとえばスレッドの取り消し時にメモリーが解放されないなど) のために、デッドロックが生じる可能性があります。pthread_cancel(3C) を使用するアプリケーションはすべて、取り消し安全環境で実行されるべきです。取り消しポイントを持ち、ロックなどのリソースを取得したり動的にメモリーを割り当てたりするライブラリは、これらのライブラリにリンクされたアプリケーションの取り消し非安全性の一因となります。これにより、マルチスレッドプログラムでのライブラリの安全性に別のレベルが導入されることになります。 取り消し安全性です。取り消し安全性には、2 つのサブカテゴリがあります。 遅延取り消し安全性、および非同期取り消し安全性です。取り消しのタイプがPTHREAD_CANCEL_DEFERRED であるスレッドが取り消し安全であるとき、アプリケーションは遅延取り消し安全であるとみなされます。取り消しのタイプがPTHREAD_CANCEL_ASYNCHRONOUS であるスレッドが、取り消し安全であるとき、アプリケーションは非同期取り消し安全であるとみなされます。非同期取り消しタイプはどこでも取り消しができる一方、遅延取り消しタイプのスレッドは十分に定義された取り消しポイントによってのみ取り消しができるため、非同期取り消し安全性よりも遅延取り消し安全性の方が目的を達成するのが容易です。すべてのスレッドはデフォルトで遅延取り消しタイプを含んで作成されるため、非同期の取り消しを安全に行うことを心配する必要はないかもしれません。ほとんどのアプリケーションおよびライブラリは、常に非同期取り消し非安全であるものと想定されています。非同期取り消し安全であるアプリケーションは、定義上は、遅延取り消し安全でもあります。

標準

多くのインタフェースが業界標準として定義され管理されています。これに該当する場合、管理している組織やバージョン管理された公開ドキュメントは、このセクションに記述されています。

移植性のあるアプリケーションを作成するプログラマは、公的標準に基づくインタフェースのマニュアルページの説明ではなく、アプリケーションが準拠しようとする標準または仕様に存在するインタフェースの説明を利用するべきです。標準または仕様に代替の実装の選択肢がある場合、通常、マニュアルページにはOracle Solaris による代替の実装だけが説明されています。マニュアルページでは、Oracle Solaris が提供する標準インタフェースの基本定義に対する互換性のある拡張についても説明されています。

管理している組織やドキュメントが「標準」エントリとして引用されている場合でも、承認されているとは見なさないでください。管理している組織には、ISO や ANSII のように非常に公的な組織もあれば、それほど公式ではないが一般に認められている IETF などの組織や、FOSS (フリーオープンソースソフトウェア) の単独貢献者のように非公式なものもあります。

関連項目

uname(1), Intro(3), standards(5)

pkg(1)

名前

audit_binfile – Solaris 監査ログの生成

形式

/usr/lib/security/audit_binfile.so

機能説明

Solaris 監査の audit_binfile プラグインモジュール (/usr/lib/security/audit_binfile.so) は、auditconfig(1M) で構成されたとおりに、バイナリ形式の監査データをファイルに書き込みます。これは、Solaris 監査デーモン auditd(1M) 用のデフォルトプラグインです。この出力は audit.log(4) によって書き込まれます。

audit_binfile プラグインが auditconfig によってアクティブとして構成されている場合は、auditd によってロードされます。プラグイン関連のすべての構成パラメータを変更するには、auditconfig -setplugin オプションを使用します。

オブジェクト属性

次の属性は、audit_binfile プラグインの構成を指定します:

p_dir

dir1[,dir2],.. [,dirn]

監査ファイルが作成されるディレクトリのリストです。有効な書き込み可能ディレクトリを指定できます。

p_minfree

ターゲット p_dir で必要な空き領域を示す割合です。空き領域がこのしきい値を下回ると、監査デーモン auditd(1M) によって、シェルスクリプト audit_warn(1M) が呼び出されます。しきい値が指定されなかった場合、デフォルトは 1% になります。

p_fsize

p_fsize 属性は、監査ファイルが自動的に閉じられ、新しい監査ファイルが開かれるまでの最大サイズを定義します。これは、監査ファイルのサイズが管理者によって指定された値と等しい場合に管理者が audit -n コマンドを発行することと同等です。デフォルトのサイズはゼロ (0) で、ファイルは無制限に増やすことができます。指定された値は、500KB よりも大きく、16 エクサバイト (EB) よりも小さくする必要があります。使用されるファイルシステムは、制限値よりも大幅に小さくなる場合があります。p_fsize 値の形式は、正確な値としてバイト単位で指定することも、B、K、M、G、T、P、E、Z (それぞれ、バイト、キロバイト、メガバイト、ギガバイト、テラバイト、ペタバイト、エクサバイト、またはゼタバイトを表す) のサフィックスを使って人間の読み取り可能な形式で指定することもできます。KB、MB、GB、TB、PB、EB、および ZB のサフィックスも使用できます。

使用例

次の指令では、audit_binfile.so がロードされ、監査ログが書き込まれるディレクトリが指定され、ディレクトリごとに必要な空き領域の割合が指定されます。

```
auditconfig -setplugin audit_binfile active \
    "p_dir=/var/audit/jedgar/eggplant,/var/audit/jedgar.aux/eggplant,
     /var/audit/global/eggplant;p_minfree=20;p_fsize=4.5GB"
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
MT レベル	MT-安全
インタフェースの安定性	確実

関連項目

auditconfig(1M)、auditd(1M)、audit_warn(1M)、syslog.conf(4)、attributes(5)

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

名前

audit_flags – 監査の事前選択フラグ

機能説明

監査フラグは、プロセスで監査される監査クラスを指定します。監査クラスは audit_class(4) ファイルで定義され、audit_event(4) ファイルで定義されたとおりに監査イベントなどをグループ化します。auditconfig(1M) を使用すると、デフォルトの Solaris システム全体の監査フラグが監査サービスの一部として構成されます。追加のユーザーごとの監査フラグまたは役割ごとの監査フラグが、user_attr(4) データベース内または *audit_flags=always-audit-flags:never-audit-flags* キーワードでユーザーに付与されたプロファイル内に構成されている場合があります。プロセスの監査フラグは事前選択マスクと呼ばれています。デフォルトの Solaris システム全体の監査フラグとユーザーごとの監査フラグ(デフォルトのフラグ + *always-audit-flags*) - *never-audit-flags* を組み合わせると、事前選択マスクがログイン時および役割の引き受け時に設定されます。

監査フラグは、監査対象の監査クラス名を表す文字列として指定されます。各フラグは監査クラスを識別し、文字列内のほかの文字とコンマ (,) で区切られます。監査クラス名の前に - が付いている場合は、失敗したクラスのみが監査対象となり、成功した試行は監査されません。監査クラス名の前に + が付いている場合は、成功したクラスのみが監査対象となり、失敗した試行は監査されません。プレフィックスが付いていない監査クラス名は、成功と失敗の両方のクラスが監査対象となることを示します。特殊な文字列 “all” を指定すると、すべての監査イベントが監査対象となります。-all を指定すると、失敗したすべての試行が監査対象となり、+all を指定すると、成功したすべての試行が監査対象となります。プレフィックス ^、^-、および ^+ は、以前に文字列に指定されたフラグを無効にします (^- は失敗の試行、^+ 成功の試行、^ は両方の試行に指定されます)。通常は、フラグをリセットするために使用します。特殊な文字列 no を指定すると、監査イベントは監査されません。

使用例

例 1 成功および失敗した「lo」(ログイン/ログアウト)、「am」(管理) 監査イベント、および失敗した「fm」(ファイル属性の変更) イベントを除くすべての失敗した監査イベントを監査する場合に事前選択します。

```
lo,am,-all,^-fm
```

例 2 成功および失敗した「lo」(ログイン/ログアウト)、「as」(システム全体の管理)、および失敗した「fm」(ファイル属性の変更) イベントを監査する場合に事前選択します。

```
lo,as,-fm
```

関連項目

profiles(1)、auditconfig(1M)、auditd(1M)、usermod(1M)、audit_class(4)、audit_event(4

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

名前　audit_remote – リモートサーバーへの Solaris 監査ログの送信

形式　/usr/lib/security/audit_remote.so

機能説明　Solaris 監査の audit_remote プラグインモジュール (/usr/lib/security/audit_remote.so) は、auditconfig(1M) で構成されたとおりに、バイナリ形式の監査レコード (audit.log(4)) を監査サーバーに送信します。

audit_remote プラグインが auditconfig によってアクティブとして構成されている場合は、auditd(1M) によってロードされます。プラグイン関連のすべての構成パラメータを変更するには、auditconfig -setplugin オプションを使用します。

audit_remote で送信されるバイナリ形式の監査レコードを受信するように、auditconfig を使用して Solaris 監査サービスデーモンの監査リモートサービス (ars(5)) が構成されている場合があります。

オブジェクト属性　次の属性は、audit_remote プラグインの構成を指定します:

p_hosts

host1[:[*port1*][:*mech1*]][,*host2*[:[*port2*][:*mech2*]],... \
hostn[:[*portn*][:*mechn*]]]

監査ホストおよびサーバーのリストです。監査レコードは、最初に到達可能なホストに送信されます。ホストに到達できない場合や、データの送信中にタイムアウトが発生した場合は、リスト内の次のホストが試されます。すべてのホストへの接続に失敗した場合は、リストの先頭から再試行されます。

p_hosts エントリの *host* 部分は、getipnodebyname(3SOCKET) に受け入れられる任意の形式で指定できます。

p_hosts エントリの *port* 部分は、監査サーバー接続を開始するためにアクセスされるホスト上のポートです。指定しない場合は、solaris-audit サービスに割り当てられたポート番号になります。getservbyname(3XNET) を参照してください。

p_host エントリの mech 部分は GSS-API メカニズム名 (mech(4)) です。指定しない場合は、ローカルホストのデフォルトメカニズムが使用されます。推奨されるメカニズムは kerberos_v5 です。

p_retries
サーバーに接続してデータを送信する再試行の回数です。

デフォルト値は 3 です。

p_timeout
接続およびデータ送信がタイムアウトになるまでの秒数です。

デフォルト値は 5 秒です。

qsize
未処理の監査レコードが保持される最大数です。

デフォルト値は、カーネルキュー制御の高位境界値です。auditconfig(1M) を参照してください。

0 に設定した場合、デフォルト値はカーネルキュー制御の高位境界値になります。auditconfig(1M) を参照してください。

GSS SESSION

audit_remote plugin は、構成された監査サーバーを GSS-API (libgss(3LIB)) を使用して認証する TCP クライアントです。Binary Solaris Audit レコードは、gss_wrap(3GSS) でメッセージトークンが生成されるたびに、完全性と機密性が保護された状態で送信されます。

このプラグインは、適切なセキュリティーメカニズム (mech(4)) を使用して、監査サーバー (*host:port:mech*) への TCP接続を開始し、GSS のセキュリティーコンテキスト (gss_init_sec_context(3GSS)) を確立します。

ポートが指定されなかった場合は、TCP ポート番号を取得するためにサービス名 solaris-audit が検索されます。メカニズムが指定されなかった場合は、GSS_C_NO_OID が gss_init_sec_context(3GSS) の mech_type パラメータとして使用されるため、ベースとなる GSS-API ではローカルのデフォルトメカニズムが使用されます。

gss_init_sec_context(3GSS) では、イニシエータの資格ハンドルとして GSS_C_NO_CREDENTIAL が使用され、audit@<host_fqdn> 形式のターゲット名が使用されます。サーバーは gss_accept_sec_context(3GSS) を使用して、コンテキストの確立を完了することが想定されています。

セキュリティーコンテキストが確立されると、クライアント (audit_remote) は gss_wrap(3GSS) を呼び出し、転送済みのペイロードと監査レコードとの機密性を保ちます。サーバーは gss_unwrap(3GSS) を使用して、受信されたデータのラップを解除し、gss_get_mic(3GSS) を使用して、あとでメッセージ取得の肯定応答としてプラグインに返信される MIC (Message Integrity Code) を取得することが想定されています。

たとえば、kerberos_v5 メカニズムがクライアント上の GSS_API メカニズムとして構成され、どちらの側もこのメカニズムを使用することに同意している場合は、クライアント側が audit/<host_fqdn>@<REALM> 主体のセッションキーを Kerberos KDC/TGS から非対話形式で取得できる必要があります。同時に、監査サーバーアプリケーションを実行しているアイデンティティーは、keytab ファイル (krb5.conf(4)) に格納された audit/<host_fqdn>@<REALM> 主体に関連付けられた長期間のキーでセッションキーを復号化できる必要があります。

audit_remote プラグインは、p_hosts リスト内の 1 番目のサーバーへの接続を開始します。接続に失敗した場合、または p_timeout 秒以内に監査レコードの送信が応答しない場合は、p_retries 回の試行後に、プラグインは次のサーバーへの接続を試みます。最後のサーバーへの接続に失敗すると、プラグインはリストの最初のホストへの接続を再度試します。サーバーへの接続の試行が失敗するたび

に、audit_warn(1M) が実行されるか、プラグインオプション audit_remote.so retry <count> <error> が付いたタイムアウトを送信します。<error> は接続 <host:port> <the network error> です。EPROTO ネットワークエラーは、クライアントプラグインがプロトコルバージョンのハンドシェイクに成功しなかったことを示します。

プロトコルの説明

すべてのプロトコルメッセージの先頭には、4 オクテットのサイズのデータが付加されています。このサイズは、ネットワークバイト順です。

このプロトコルではバージョンのネゴシエーションで始まり、GSS-API セキュリティーコンテキストトークンの交換が続きます。エラー発生時には、接続が閉じられます (オプションで出力トークンが送信されます)。

バージョンのネゴシエーションは、サポートされるバージョンをコンマ (,) で区切ったリストのオクテット配列を送信するプラグインを使用して、平文で実行されます。現在のバージョン番号は文字 01 です。受信側は受け入れたバージョンで応答することが想定されています (現在の場合は文字 01)。不一致はエラーとみなされ、接続が閉じられます。

プラグインで送信されるバージョンのオクテット配列と受信側で受け入れられるバージョン文字を互いに連結すると、GSS セキュリティーコンテキスト確立のチャネルバインディングのアプリケーションデータフィールドが形成されます。

```
<plugin version characters> || <server accepted version characters>
"||" represents concatenation
```

以降のトークンには、ネットワークバイトオーダーで並んだ 64 ビットのシーケンス番号と単一の監査レコード (audit.log(4)) が含まれます。クライアントは機密性保護ラップ (64 ビットのシーケンス番号 || 監査レコード) を使用します。

サーバーは、受信した 64 ビットのシーケンス番号、ラップされていない 64 ビットのシーケンス番号が付いた MIC トークン、および監査レコードによって、受領したことを肯定応答します (その後、データ損失の対応をします)。クライアント側の MIC 検証によって、監査レコードは解放できるが、再転送のために保存できないことが肯定応答されます。

```
64 bit sequence number || mic (64 bit sequence number || audit record)
```

セキュアなリモート監査クライアントとサーバー間の通信フロー:

```
1) Client <--> Server - TCP handshake

2) Client <--> Server - protocol version negotiation:
   a) Client  --> Server - send data size - uint32_t value (2)
   b) Client  --> Server - send clear text message of the versions
                           supported comma separated, e.g.,
                           "01,02,03" for versions 1 and 2 and 3.
                           The only version supported at present is
                           "01"
```

```
   c) Client <--  Server - send data size - uint32_t value (2)
   d) Client <--  Server - send clear text version selected
                           ("01")
   :no version match; close connection; try next host

3) Security context initiation:
   a) Client - Construct channel bindings:
         initiator address type (GSS_C_AF_NULLADDR)
         acceptor address type (GSS_C_AF_NULLADDR)
         application data value (4 octets  "0101")
   b) Client  --> Server - send token (data) size - uint32_t value
   c) Client  --> Server - GSS-API per-context token
   d) Client <--  Server - send token (data) size
   e) Client <--  Server - GSS-API per-context token
      :repeat a-e until security context is initialized; if unsuccessful,
      close connection; try next host

4) Client - transmit thread, when audit record to be sent:
   a) Client  --> Server - send data size
   b) Client  --> Server - GSS-API per-message token
                  wrap (sequence number || audit record)
      :repeat a-b while less than max (qsize) outstanding records

 5) Client - receive thread:
    a) Client <--  Server - receive data size - uint32_t value
    b) Client <--  Server - receive sequence number - uint64_t value
    c) Client <--  Server - receive MIC
    d) Client             - MIC verification - OK
    e) Client             - remove particular audit record
                            pointed by the sequence number from the
                            retransmit buffer
  :repeat a-e, on error close connection; try next host;
   retransmit unacknowledged audit records

6) Server - receive thread:
    a) Client  --> Server - receive data size
    b) Client  --> Server - GSS-API receive, uwrap, store
                   per-message token

7) Server - transmit thread:
    a) Server - MIC generation - message integrity code
                    mic (sequence number || audit record)
    b) Client <--  Server - send data size
    c) Client < -- Server - send sequence number
    d) Client <--  Server - send MIC
```

使用例

例 1 audit_remote.so のロードとリモート監査サーバーの指定

次の指令では、audit_remote.so がロードされ、監査レコードが送信されるリモート監査サーバーが指定されます。kerberos_v5 セキュリティーメカニズムは、サーバーとの通信時に使用されるように定義されています。

```
auditconfig -setplugin audit_remote active \
    "p_timeout=90;p_retries=2;
     p_hosts=eggplant.eng.sun.com::kerberos_v5,
     purple.ebay.sun.com:4592:kerberos_v5"
```

例 2 デフォルトのセキュリティーメカニズムの使用法の構成

次の例では、デフォルトのセキュリティーメカニズムの使用法の構成を示します。構成されたサーバーのいずれかのデフォルトポートの使用法も示します:

```
auditconfig -setplugin audit_remote active \
    "p_timeout=10;p_retries=2;
     p_hosts=jedger.eng.sun.com, jbadams.ebay.sun.com:4592"
```

例 3 内部プラグインのキューサイズの設定

特定の状況 (たとえば、サーバーとクライアント間の通信回線が低速になることに加えて、監査データトラフィック量が増加したり、バーストしたりする状況) では、audit_remote プラグインによってキューに入れられた未処理の監査レコードの数が、構成された最大数に到達する可能性があります。次の例では、キューサイズのパラメータを設定する方法を示します。

```
auditconfig -setplugin audit_remote "" 1000
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
MT レベル	MT-安全
インタフェースの安定性	下記を参照。

プラグイン構成パラメータは「確実」です。クライアント/サーバープロトコル (バージョン "01") は「契約済みプロジェクト非公開」です。監査レコードの形式と内容の安定性については、audit.log(4) を参照してください。

関連項目

auditd(1M)、auditconfig(1M)、audit_warn(1M)、getipnodebyname(3SOCKET)、getservbyname(3X

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

注意事項

audit_remote は、GSS-API (libgss(3LIB)) を使用して、リモート監査サービスに対する認証を行います。Kerberos などの gss 実装メカニズムで提供されるデフォルトの gss 資格情報が使用されます。

IANA で割り当てられた `solaris-audit` サービスポートは `16162` です。

名前

audit_syslog – syslog メッセージへの Solaris 監査データのリアルタイム変換

形式

```
/usr/lib/security/audit_syslog.so
```

機能説明

Solaris 監査の audit_syslog プラグインモジュール (/usr/lib/security/audit_syslog.so) は、syslog.conf(4) で構成されたとおりに、Solaris 監査データを syslog 形式の (テキスト) データにリアルタイムで変換し、syslog デーモンに送信します。プラグインのパスは、auditconfig(1M) ユーティリティーを使用して指定されます。

plugin が auditconfig によってアクティブとして構成されている場合は、syslog にメッセージが書き込まれます。プラグイン関連のすべての構成パラメータを変更するには、auditconfig -setplugin オプションを使用します。syslog メッセージは、LOG_AUDIT (syslog.conf(4) の audit) の機能コードと LOG_NOTICE の重要度を使用して生成されます。監査 syslog メッセージには、バイナリ形式の監査ログ用に記述されたトークンから選択したデータが含まれます。(audit.log(4) を参照)。すべての syslog メッセージと同様に、syslog ファイルの各行は syslog ヘッダーとメッセージの 2 つの部分で構成されます。

syslog ヘッダーには、メッセージが生成された日付と時間、送信元のホスト名、監査デーモンによって生成されたことを示す auditd、syslog で内部的に使用される ID フィールド、および syslogd 機能と重要度の値を示す audit.notice が含まれます。syslog ヘッダーは文字] (すなわち、終了角括弧とスペース) で終わります。

メッセージ部分は、ヘッダートークンから生成されたイベントタイプで始まります。以降のすべてのデータは、元の監査レコードに含まれ、1024 バイトの最大長の syslog 行に余裕がある場合にのみ表示されます。次の例では、バックスラッシュ文字 (\) は継続することを示しています。つまり、実際の syslog メッセージは 1 行で記述されています:

```
Oct 31 11:38:08 smothers auditd: [ID 917521 audit.notice] chdir(2) ok\
session 401 by joeuser as root:other from myultra obj /export/home
```

前述の例では、chdir(2) はイベントタイプです。このフィールド以降は、次に説明する追加データです。このデータはソースの監査レコードに含まれていない場合、省略されます。

ok または failed	return または exit トークンから生成されます。
session <#>	<#> は subject トークンから生成されたセッション ID です。
by <*name*>	<*name*> は subject トークンから生成された監査 ID です。
as <*name*>:<*group*>	<*name*> は subject トークンから生成された実効ユーザー ID、<*group*> は実行グループ ID です。
in <*zone name*>	ゾーン名。このフィールドは、zonename 監査ポリシーが設定されている場合にのみ生成されます。

`from` <*terminal*>	<*terminal*> は subject トークンから生成されたテキストのマシンアドレスです。
`obj` <*path*>	<*path*> は path トークンから生成されたパスです。必要に応じて、パスを左側から切り捨て、行に収めることができます。切り捨ては先行する省略符号 (. . .) で示されます。
`proc_uid` <*owner*>	<*owner*> はプロセス所有者の実効ユーザー ID です。
`proc_auid` <*owner*>	<*owner*> はプロセス所有者の監査 ID です。

次に、`syslog` メッセージの例を示します:

```
Nov  4  8:27:07 smothers auditd: [ID 175219 audit.notice]
\system booted

Nov  4  9:28:17 smothers auditd: [ID 752191 audit.notice] \
login - rlogin ok session 401 by joeuser as joeuser:staff from myultra

Nov  4 10:29:27 smothers auditd: [ID 521917 audit.notice] \
access(2) ok session 255 by janeuser as janeuser:staff from  \
129.146.89.30 obj /etc/passwd
```

オブジェクト属性

`p_flag` 属性は、`syslog` デーモンに送信される監査データを、`flags` と `naflags` (auditconfig(1M) を参照) および user_attr(4) のユーザー固有の行から指定されたクラスよりもさらに絞り込むために使用されます。このパラメータはコンマで区切ったリストです。各項目は監査クラス (audit_class(4) を参照) を表し、audit_flags(5) で説明されている構文を使用して指定されます。デフォルト (空の `p_flags` が一覧表示される) では、監査レコードは生成されません。

使用例

例 1　plugin 行の使用例

次に示す指定では、`plugin` (`flags` と `naflags` の設定と組み合わせて) を使用すると、`lo` のクラスレコードを使用できますが、`am` のクラスレコードは失敗した場合にしか使用できません。`fm` クラスレコードを省略すると、`fm` クラスレコードが出力されません。`flags` と `naflags` および user_attr(4) を使用して定義されたものにはクラスを追加できないため、`pc` パラメータは影響を受けません。削除のみ可能です。

```
auditconfig -setflags lo,am,fm
auditconfig -setnaflags lo
auditconfig -setplugin audit_syslog active "p_flags=lo,-am,pc"
```

例 2　all の使用法

次に示す指定では、`all` では、1 つの例外を除いて、`flags` と `naflags` (および user_attr(4)) を使用して定義されたすべてのフラグを使用できます。`am` メタクラス (`ss,as,ua` と同等) は、すべての `ua` イベント (ただし、`ss` と `as` は失敗イベントのみ) を出力するように変更されているという例外があります。

例 2 allの使用法 (続き)

```
auditconfig -setflags lo,am
auditconfig -setnaflags lo
auditconfig -setplugin audit_syslog active "p_flags=all,^+ss,^+as"
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
MT レベル	MT-安全
インタフェースの安定性	下記を参照。

メッセージの形式および内容は「不確実」です。構成パラメータは「確実」です。

関連項目

auditconfig(1M)、auditd(1M)、audit_class(4)、syslog.conf(4)、user_attr(4)、attributes(5)、

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

注意事項

audit_syslog plugin をアクティブ化するには、audit 機能および notice 重要度の syslog メッセージ、または Solaris 監査レコード用の前述のファイル内の syslog メッセージが格納されるように /etc/syslog.conf が構成されている必要があります。次に、syslog.conf 内のこのような行の例を示します:

```
audit.notice                    /var/audit/audit.log
```

syslog からのメッセージは、UDP を使用してリモートの syslog サーバーに送信されるため、配信が保証されなかったり、メッセージの正確な到達順序が保証されなかったりします。

plugin 行に指定されたパラメータによってクラスが事前選択されていない場合は、LOG_DAEMON 機能コードが含まれる syslog アラートを使用してエラーが報告されます。

syslog ヘッダーの時間フィールドは syslog(3C) によって生成され、バイナリ形式の監査ログにはだいたいの時間しか書き込まれません。通常は、時間フィールドには、1 秒か数秒程度の違いで表示されます。

名前

brands – 非大域ゾーンのオペレーティング環境の切り替え

機能説明

ブランドゾーン (BrandZ) フレームワークは、zones(5) で説明されている Solaris ゾーンインフラストラクチャーを拡張して、ブランドの作成機能を追加します。ブランドは、ネイティブでないオペレーティング環境を含む非大域ゾーンを提供します。

「ブランド」という用語は、さまざまなオペレーティング環境を指す場合があります。すべてのブランド管理は、現在のゾーン構造の拡張を通して実行されます。

すべてのゾーンに、それぞれ関連するブランドが構成されます。ブランドタイプにより、ゾーンのインストール時およびブート時に実行されるスクリプトが決定されます。また、ゾーンのブランドにより、アプリケーションの起動時に正しいアプリケーションタイプが適切に識別されます。デフォルトのブランドは、大域ゾーンにインストールされているディストリビューションによって決まります。

ブランドゾーンは、ネイティブでないバイナリのブランドを 1 つだけサポートします。つまり、1 つのブランドゾーンは 1 つのオペレーティング環境を提供します。ゾーンにブランドを割り当てたあとは、そのブランドの変更や削除を行うことはできません。

BrandZ はゾーンのツールを次のように拡張します。

- ブランドはゾーンの属性であり、ゾーンの作成時に設定されます。
- zonecfg ツール (zonecfg(1M) を参照) を使用して、ゾーンのブランドタイプの設定とゾーンの構成を行います。
- zoneadm ツール (zoneadm(1M) を参照) を使用して、ゾーンのブランドタイプの報告とゾーンの管理を行います。

ブランド

Oracle Solaris 11 では、利用可能なブランドは次のとおりです。

- `solaris`

  solaris(5) を参照してください。

- `solaris10`

  solaris10(5) を参照してください。

- `labeled`

  Trusted Extensions を実装します。txzonemgr(1M) を参照してください。

デバイスのサポート

各ゾーンでサポートされるデバイスについては、そのブランドに関するマニュアルページやほかのドキュメントに記載されています。サポートされていないデバイスを追加しようとすると、ゾーンインフラストラクチャーで検出され、管理者に警告が発行されます。管理者が警告にもかかわらず、サポートされていないデバイスを追加した場合、そのデバイスが期待どおりに機能するかどうかは不明です。その構成はテストもサポートもされません。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	確実

関連項目

mdb(1), zlogin(1), zonename(1), dtrace(1M), in.rlogind(1M), sshd(1M), txzonemgr(1M), zoneadm(1M), zonecfg(1M), kill(2), priocntl(2), getzoneid(3C), ucred_get(3C), getzoneid(3C), proc(4), attributes(5), solaris(5), privileges(5), solaris(5), solaris10(5), zones(5), crgetzoneid(9F)

名前

cancellation – overview of concepts related to POSIX thread cancellation

機能説明

FUNCTION	ACTION
`pthread_cancel()`	Cancels thread execution.
`pthread_setcancelstate()`	Sets the cancellation *state* of a thread.
`pthread_setcanceltype()`	Sets the cancellation *type* of a thread.
`pthread_testcancel()`	Creates a cancellation point in the calling thread.
`pthread_cleanup_push()`	Pushes a cleanup handler routine.
`pthread_cleanup_pop()`	Pops a cleanup handler routine.

Cancellation

Thread cancellation allows a thread to terminate the execution of any application thread in the process. Cancellation is useful when further operations of one or more threads are undesirable or unnecessary.

An example of a situation that could benefit from using cancellation is an asynchronously-generated cancel condition such as a user requesting to close or exit some running operation. Another example is the completion of a task undertaken by a number of threads, such as solving a maze. While many threads search for the solution, one of the threads might solve the puzzle while the others continue to operate. Since they are serving no purpose at that point, they should all be canceled.

Planning Steps

Planning and programming for most cancellations follow this pattern:

1. Identify which threads you want to cancel, and insert `pthread_cancel(3C)` statements.
2. Identify system-defined cancellation points where a thread that might be canceled could have changed system or program state that should be restored. See the `Cancellation Points` for a list.
3. When a thread changes the system or program state just before a cancellation point, and should restore that state before the thread is canceled, place a cleanup handler before the cancellation point with `pthread_cleanup_push(3C)`. Wherever a thread restores the changed state, pop the cleanup handler from the cleanup stack with `pthread_cleanup_pop(3C)`.
4. Know whether the threads you are canceling call into cancel-unsafe libraries, and disable cancellation with `pthread_setcancelstate(3C)` before the call into the library. See `Cancellation State` and `Cancel-Safe`.
5. To cancel a thread in a procedure that contains no cancellation points, insert your own cancellation points with `pthread_testcancel(3C)`. This function creates cancellation points by testing for pending cancellations and performing those cancellations if they are found. Push and pop cleanup handlers around the cancellation point, if necessary (see Step 3, above).

Cancellation Points

The system defines certain points at which cancellation can occur (cancellation points), and you can create additional cancellation points in your application with `pthread_testcancel()`.

The following cancellation points are defined by the system (system-defined cancellation points): `creat(2)`, `aio_suspend(3C)`, `close(2)`, `creat(2)`, `getmsg(2)`, `getpmsg(2)`, `lockf(3C)`, `mq_receive(3C)`, `mq_send(3C)`, `msgrcv(2)`, `msgsnd(2)`, `msync(3C)`, `nanosleep(3C)`, `open(2)`, `pause(2)`, `poll(2)`, `pread(2)`, `pthread_cond_timedwait(3C)`, `pthread_cond_wait(3C)`, `pthread_join(3C)`, `pthread_testcancel(3C)`, `putmsg(2)`, `putpmsg(2)`, `pwrite(2)`, `read(2)`, `readv(2)`, `select(3C)`, `sem_wait(3C)`, `sigpause(3C)`, `sigwaitinfo(3C)`, `sigsuspend(2)`, `sigtimedwait(3C)`, `sigwait(2)`, `sleep(3C)`, `sync(2)`, `system(3C)`, `tcdrain(3C)`, `usleep(3C)`, `wait(3C)`, `waitid(2)`, `wait3(3C)`, `waitpid(3C)`, `write(2)`, `writev(2)`, and `fcntl(2)`, when specifying `F_SETLKW` as the command.

When cancellation is asynchronous, cancellation can occur at any time (before, during, or after the execution of the function defined as the cancellation point). When cancellation is deferred (the default case), cancellation occurs only within the scope of a function defined as a cancellation point (after the function is called and before the function returns). See `Cancellation Type` for more information about deferred and asynchronous cancellation.

Choosing where to place cancellation points and understanding how cancellation affects your program depend upon your understanding of both your application and of cancellation mechanics.

Typically, any call that might require a long wait should be a cancellation point. Operations need to check for pending cancellation requests when the operation is about to block indefinitely. This includes threads waiting in `pthread_cond_wait()` and `pthread_cond_timedwait()`, threads waiting for the termination of another thread in `pthread_join()`, and threads blocked on `sigwait()`.

A mutex is explicitly not a cancellation point and should be held for only the minimal essential time.

Most of the dangers in performing cancellations deal with properly restoring invariants and freeing shared resources. For example, a carelessly canceled thread might leave a mutex in a locked state, leading to a deadlock. Or it might leave a region of memory allocated with no way to identify it and therefore no way to free it.

Cleanup Handlers

When a thread is canceled, it should release resources and clean up the state that is shared with other threads. So, whenever a thread that might be canceled changes the state of the system or of the program, be sure to push a cleanup handler with `pthread_cleanup_push(3C)` before the cancellation point.

When a thread is canceled, all the currently-stacked cleanup handlers are executed in last-in-first-out (LIFO) order. Each handler is run in the scope in which it was pushed. When the last cleanup handler returns, the thread-specific data destructor functions are called. Thread execution terminates when the last destructor function returns.

When, in the normal course of the program, an uncanceled thread restores state that it had previously changed, be sure to pop the cleanup handler (that you had set up where the change took place) using `pthread_cleanup_pop(3C)`. That way, if the thread is canceled later, only currently-changed state will be restored by the handlers that are left in the stack.

The `pthread_cleanup_push()` and `pthread_cleanup_pop()` functions can be implemented as macros. The application must ensure that they appear as statements, and in pairs within the same lexical scope (that is, the `pthread_cleanup_push()` macro can be thought to expand to a token list whose first token is '{' with `pthread_cleanup_pop()` expanding to a token list whose last token is the corresponding '}').

The effect of the use of `return`, `break`, `continue`, and `goto` to prematurely leave a code block described by a pair of `pthread_cleanup_push()` and `pthread_cleanup_pop()` function calls is undefined.

Cancellation State

Most programmers will use only the default cancellation state of `PTHREAD_CANCEL_ENABLE`, but can choose to change the state by using `pthread_setcancelstate(3C)`, which determines whether a thread is cancelable at all. With the default *state* of `PTHREAD_CANCEL_ENABLE`, cancellation is enabled and the thread is cancelable at points determined by its cancellation *type*. See `Cancellation Type`.

If the *state* is `PTHREAD_CANCEL_DISABLE`, cancellation is disabled, the thread is not cancelable at any point, and all cancellation requests to it are held pending.

You might want to disable cancellation before a call to a cancel-unsafe library, restoring the old cancel state when the call returns from the library. See `Cancel-Safe` for explanations of cancel safety.

Cancellation Type

A thread's cancellation `type` is set with `pthread_setcanceltype(3C)`, and determines whether the thread can be canceled anywhere in its execution or only at cancellation points.

With the default *type* of `PTHREAD_CANCEL_DEFERRED`, the thread is cancelable only at cancellation points, and then only when cancellation is enabled.

If the *type* is `PTHREAD_CANCEL_ASYNCHRONOUS`, the thread is cancelable at any point in its execution (assuming, of course, that cancellation is enabled). Try to limit regions of asynchronous cancellation to sequences with no external dependencies that could result in dangling resources or unresolved state conditions. Using asynchronous cancellation is discouraged because of the danger involved in trying to guarantee correct cleanup handling at absolutely every point in the program.

Cancellation Type/State Table

Type	State	
	Enabled (Default)	Disabled

Cancellation Type/State Table		
Deferred (Default)	Cancellation occurs when the target thread reaches a cancellation point and a cancel is pending. (Default)	All cancellation requests to the target thread are held pending.
Asynchronous	Receipt of a `pthread_cancel()` call causes immediate cancellation.	All cancellation requests to the target thread are held pending; as soon as cancellation is re-enabled, pending cancellations are executedimmediately.

Cancel-Safe

With the arrival of POSIX cancellation, the Cancel-Safe level has been added to the list of MT-Safety levels. See attributes(5). An application or library is Cancel-Safe whenever it has arranged for cleanup handlers to restore system or program state wherever cancellation can occur. The application or library is specifically Deferred-Cancel-Safe when it is Cancel-Safe for threads whose cancellation type is `PTHREAD_CANCEL_DEFERRED`. See `Cancellation State`. It is specifically Asynchronous-Cancel-Safe when it is Cancel-Safe for threads whose cancellation type is `PTHREAD_CANCEL_ASYNCHRONOUS`.

It is easier to arrange for deferred cancel safety, as this requires system and program state protection only around cancellation points. In general, expect that most applications and libraries are not Asynchronous-Cancel-Safe.

POSIX Threads Only

The cancellation functions described in this manual page are available for POSIX threads, only (the Solaris threads interfaces do not provide cancellation functions).

使用例

例 1 Cancellation example

The following short C++ example shows the pushing/popping of cancellation handlers, the disabling/enabling of cancellation, the use of `pthread_testcancel()`, and so on. The `free_res()` cancellation handler in this example is a dummy function that simply prints a message, but that would free resources in a real application. The function `f2()` is called from the main thread, and goes deep into its call stack by calling itself recursively.

Before `f2()` starts running, the newly created thread has probably posted a cancellation on the main thread since the main thread calls `thr_yield()` right after creating thread2. Because cancellation was initially disabled in the main thread, through a call to `pthread_setcancelstate()`, the call to `f2()` from `main()` continues and constructs X at each recursive call, even though the main thread has a pending cancellation.

When `f2()` is called for the fifty-first time (when "`i == 50`"), `f2()` enables cancellation by calling `pthread_setcancelstate()`. It then establishes a cancellation point for itself by calling `pthread_testcancel()`. (Because a cancellation is pending, a call to a cancellation point such as read(2) or write(2) would also cancel the caller here.)

例 1 Cancellation example *(続き)*

After the `main()` thread is canceled at the fifty-first iteration, all the cleanup handlers that were pushed are called in sequence; this is indicated by the calls to `free_res()` and the calls to the destructor for *X*. At each level, the C++ runtime calls the destructor for *X* and then the cancellation handler, `free_res()`. The print messages from `free_res()` and *X*'s destructor show the sequence of calls.

At the end, the main thread is joined by thread2. Because the main thread was canceled, its return status from `pthread_join()` is `PTHREAD_CANCELED`. After the status is printed, thread2 returns, killing the process (since it is the last thread in the process).

```
#include <pthread.h>
#include <sched.h>
extern "C" void thr_yield(void);

extern "C" void printf(...);

struct X {
        int x;
        X(int i){x = i; printf("X(%d) constructed.\n", i);}
        ~X(){ printf("X(%d) destroyed.\n", x);}
};

void
free_res(void *i)
{
        printf("Freeing ‘%d‘\n",i);
}

char* f2(int i)
{
        try {
        X dummy(i);
        pthread_cleanup_push(free_res, (void *)i);
        if (i == 50) {
               pthread_setcancelstate(PTHREAD_CANCEL_ENABLE, NULL);
            pthread_testcancel();
        }
        f2(i+1);
        pthread_cleanup_pop(0);
        }
        catch (int) {
        printf("Error: In handler.\n");
        }
        return "f2";
}
```

例 1 Cancellation example *(続き)*

```
void *
thread2(void *tid)
{
        void *sts;

        printf("I am new thread :%d\n", pthread_self());

        pthread_cancel((pthread_t)tid);

        pthread_join((pthread_t)tid, &sts);

        printf("main thread cancelled due to %d\n", sts);

        return (sts);
}

main()
{
        pthread_setcancelstate(PTHREAD_CANCEL_DISABLE, NULL);
        pthread_create(NULL, NULL, thread2, (void *)pthread_self());
        thr_yield();
        printf("Returned from %s\n",f2(0));
}
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe

関連項目

read(2), sigwait(2), write(2), Intro(3), condition(5), pthread_cleanup_pop(3C), pthread_cleanup_push(3C), pthread_exit(3C), pthread_join(3C), pthread_setcancelstate(3C), pthread_setcanceltype(3C), pthread_testcancel(3C), setjmp(3C), attributes(5), standards(5)

名前

charmap – character set description file

機能説明

A character set description file or *charmap* defines characteristics for a coded character set. Other information about the coded character set may also be in the file. Coded character set character values are defined using symbolic character names followed by character encoding values.

The character set description file provides:

- The capability to describe character set attributes (such as collation order or character classes) independent of character set encoding, and using only the characters in the portable character set. This makes it possible to create generic `localedef(1)` source files for all codesets that share the portable character set.
- Standardized symbolic names for all characters in the portable character set, making it possible to refer to any such character regardless of encoding.

Symbolic Names

Each symbolic name is included in the file and is mapped to a unique encoding value (except for those symbolic names that are shown with identical glyphs). If the control characters commonly associated with the symbolic names in the following table are supported by the implementation, the symbolic names and their corresponding encoding values are included in the file. Some of the encodings associated with the symbolic names in this table may be the same as characters in the portable character set table.

<ACK>	<DC2>	<ENQ>	<FS>	<IS4>	<SOH>
<BEL>	<DC3>	<EOT>	<GS>	<LF>	<STX>
<BS>	<DC4>	<ESC>	<HT>	<NAK>	<SUB>
<CAN>	<DEL>	<ETB>	<IS1>	<RS>	<SYN>
<CR>	<DLE>	<ETX>	<IS2>	<SI>	<US>
<DC1>	<EM>	<FF>	<IS3>	<SO>	<VT>

Declarations

The following declarations can precede the character definitions. Each must consist of the symbol shown in the following list, starting in column 1, including the surrounding brackets, followed by one or more blank characters, followed by the value to be assigned to the symbol.

<*code_set_name*> The name of the coded character set for which the character set description file is defined.

<*mb_cur_max*> The maximum number of bytes in a multi-byte character. This defaults to 1.

<*mb_cur_min*> An unsigned positive integer value that defines the minimum number of bytes in a character for the encoded character set.

<*escape_char*> The escape character used to indicate that the characters following will be interpreted in a special way, as defined later in this section. This defaults to backslash ('\'), which is the character glyph used in all the following text and examples, unless otherwise noted.

<*comment_char*> The character that when placed in column 1 of a charmap line, is used to indicate that the line is to be ignored. The default character is the number sign (#).

Format

The character set mapping definitions will be all the lines immediately following an identifier line containing the string `CHARMAP` starting in column 1, and preceding a trailer line containing the string `END CHARMAP` starting in column 1. Empty lines and lines containing a <*comment_char*> in the first column will be ignored. Each non-comment line of the character set mapping definition, that is, between the `CHARMAP` and `END CHARMAP` lines of the file), must be in either of two forms:

"`%s %s %s\n`",<*symbolic-name*>,<*encoding*>,<*comments*>

or

"`%s...%s %s %s\n`",<*symbolic-name*>,<*symbolic-name*>, <*encoding*>,\
<*comments*>

In the first format, the line in the character set mapping definition defines a single symbolic name and a corresponding encoding. A character following an escape character is interpreted as itself; for example, the sequence "<\\>>" represents the symbolic name "\>" enclosed between angle brackets.

In the second format, the line in the character set mapping definition defines a range of one or more symbolic names. In this form, the symbolic names must consist of zero or more non-numeric characters, followed by an integer formed by one or more decimal digits. The characters preceding the integer must be identical in the two symbolic names, and the integer formed by the digits in the second symbolic name must be equal to or greater than the integer formed by the digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers between the first and the second integer, inclusive. As an example, `<j0101>...<j0104>` is interpreted as the symbolic names `<j0101>`, `<j0102>`, `<j0103>`, and `<j0104>`, in that order.

A character set mapping definition line must exist for all symbolic names and must define the coded character value that corresponds to the character glyph indicated in the table, or the coded character value that corresponds with the control character symbolic name. If the control characters commonly associated with the symbolic names are supported by the implementation, the symbolic name and the corresponding encoding value must be included in the file. Additional unique symbolic names may be included. A coded character value can be represented by more than one symbolic name.

The encoding part is expressed as one (for single-byte character values) or more concatenated decimal, octal or hexadecimal constants in the following formats:

"%cd%d",<*escape_char*>,<*decimal byte value*>

"%cx%x",<*escape_char*>,<*hexadecimal byte value*>

"%c%o",<*escape_char*>,<*octal byte value*>

Decimal Constants

Decimal constants must be represented by two or three decimal digits, preceded by the escape character and the lower-case letter d; for example, \d05, \d97, or \d143. Hexadecimal constants must be represented by two hexadecimal digits, preceded by the escape character and the lower-case letter x; for example, \x05, \x61, or \x8f. Octal constants must be represented by two or three octal digits, preceded by the escape character; for example, \05, \141, or \217. In a portable charmap file, each constant must represent an 8-bit byte. Implementations supporting other byte sizes may allow constants to represent values larger than those that can be represented in 8-bit bytes, and to allow additional digits in constants. When constants are concatenated for multi-byte character values, they must be of the same type, and interpreted in byte order from first to last with the least significant byte of the multi-byte character specified by the last constant.

Ranges of Symbolic Names

In lines defining ranges of symbolic names, the encoded value is the value for the first symbolic name in the range (the symbolic name preceding the ellipsis). Subsequent symbolic names defined by the range will have encoding values in increasing order. Bytes are treated as unsigned octets and carry is propagated between the bytes as necessary to represent the range. However, because this causes a null byte in the second or subsequent bytes of a character, such a declaration should not be specified. For example, the line

```
<j0101>...<j0104>     \d129\d254
```

is interpreted as:

```
<j0101>                \d129\d254
<j0102>                \d129\d255
<j0103>                \d130\d00
<j0104>                \d130\d01
```

The expanded declaration of the symbol <j0103> in the above example is an invalid specification, because it contains a null byte in the second byte of a character.

The comment is optional.

Width Specification

The following declarations can follow the character set mapping definitions (after the "END CHARMAP" statement). Each consists of the keyword shown in the following list, starting in column 1, followed by the value(s) to be associated to the keyword, as defined below.

WIDTH A non-negative integer value defining the column width for the printable character in the coded character set mapping definitions. Coded

character set character values are defined using symbolic character names followed by column width values. Defining a character with more than one WIDTH produces undefined results. The END WIDTH keyword is used to terminate the WIDTH definitions. Specifying the width of a non-printable character in a WIDTH declaration produces undefined results.

WIDTH_DEFAULT — A non-negative integer value defining the default column width for any printable character not listed by one of the WIDTH keywords. If no WIDTH_DEFAULT keyword is included in the charmap, the default character width is 1.

Example:

After the "END CHARMAP" statement, a syntax for a width definition would be:

```
WIDTH
<A>             1
<B>             1
<C>...<Z>       1
...
<fool>...<foon> 2
...
END WIDTH
```

In this example, the numerical code point values represented by the symbols <A> and <B> are assigned a width of 1. The code point values <C> to <Z> inclusive, that is, <C>, <D>, <E>, and so on, are also assigned a width of 1. Using <A>. . .<Z> would have required fewer lines, but the alternative was shown to demonstrate flexibility. The keyword WIDTH_DEFAULT could have been added as appropriate.

関連項目

locale(1), localedef(1), nl_langinfo(3C), extensions(5), locale(5)

名前

condition – concepts related to condition variables

機能説明

Occasionally, a thread running within a mutex needs to wait for an event, in which case it blocks or sleeps. When a thread is waiting for another thread to communicate its disposition, it uses a condition variable in conjunction with a mutex. Although a mutex is exclusive and the code it protects is sharable (at certain moments), condition variables enable the synchronization of differing events that share a mutex, but not necessarily data. Several condition variables may be used by threads to signal each other when a task is complete, which then allows the next waiting thread to take ownership of the mutex.

A condition variable enables threads to atomically block and test the condition under the protection of a mutual exclusion lock (mutex) until the condition is satisfied. If the condition is false, a thread blocks on a condition variable and atomically releases the mutex that is waiting for the condition to change. If another thread changes the condition, it may wake up waiting threads by signaling the associated condition variable. The waiting threads, upon awakening, reacquire the mutex and re-evaluate the condition.

Initialize

Condition variables and mutexes should be global. Condition variables that are allocated in writable memory can synchronize threads among processes if they are shared by the cooperating processes (see `mmap(2)`) and are initialized for this purpose.

The scope of a condition variable is either intra-process or inter-process. This is dependent upon whether the argument is passed implicitly or explicitly to the initialization of that condition variable. A condition variable does not need to be explicitly initialized. A condition variable is initialized with all zeros, by default, and its scope is set to within the calling process. For inter-process synchronization, a condition variable must be initialized once, and only once, before use.

A condition variable must not be simultaneously initialized by multiple threads or re-initialized while in use by other threads.

Condition variables attributes may be set to the default or customized at initialization. POSIX threads even allow the default values to be customized. Establishing these attributes varies depending upon whether POSIX or Solaris threads are used. Similar to the distinctions between POSIX and Solaris thread creation, POSIX condition variables implement the default, intra-process, unless an attribute object is modified for inter-process prior to the initialization of the condition variable. Solaris condition variables also implement as the default, intra-process; however, they set this attribute according to the argument, *type*, passed to their initialization function.

Condition Wait

The condition wait interface allows a thread to wait for a condition and atomically release the associated mutex that it needs to hold to check the condition. The thread waits for another thread to make the condition true and that thread's resulting call to signal and wakeup the waiting thread.

Condition Signaling

A condition signal allows a thread to unblock the next thread waiting on the condition variable, whereas, a condition broadcast allows a thread to unblock all threads waiting on the condition variable.

Destroy

The condition destroy functions destroy any state, but not the space, associated with the condition variable.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe

関連項目

fork(2), mmap(2), setitimer(2), shmop(2), cond_broadcast(3C), cond_destroy(3C), cond_init(3C), cond_signal(3C), cond_timedwait(3C), cond_wait(3C), pthread_cond_broadcast(3C), pthread_cond_destroy(3C), pthread_cond_init(3C), pthread_cond_signal(3C), pthread_cond_timedwait(3C), pthread_cond_wait(3C), pthread_condattr_init(3C), signal(3C), attributes(5), mutex(5), standards(5)

注意事項

If more than one thread is blocked on a condition variable, the order in which threads are unblocked is determined by the scheduling policy.

USYNC_THREAD does not support multiple mapplings to the same logical synch object. If you need to mmap() a synch object to different locations within the same address space, then the synch object should be initialized as a shared object USYNC_PROCESS for Solaris, and PTHREAD_PROCESS_PRIVATE for POSIX.

名前

crypt_bsdbf – password hashing module using Blowfish cryptographic algorithm

形式

`/usr/lib/security/$ISA/crypt_bsdbf.so`

機能説明

The `crypt_bsdbf` module is a one-way password hashing module for use with `crypt(3C)` that uses the Blowfish cryptographic algorithm. The algorithm identifier for `crypt.conf(4)` and `policy.conf(4)` is `2a`.

The maximum password length for `crypt_bsdbf` is 72 characters.

The default number of rounds is `log(4)`. This can be increased by updating `/etc/security/crypt.conf`, for example, for `log(12)` rounds:

```
2a crypt_bsdbf.so.1    12
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`

名前

crypt_bsdmd5 – password hashing module using MD5 message hash algorithm

形式

`/usr/lib/security/$ISA/crypt_bsdmd5.so`

機能説明

The `crypt_bsdmd5` module is a one-way password hashing module for use with `crypt(3C)` that uses the MD5 message hash algorithm. The algorithm identifier for `crypt.conf(4)` and `policy.conf(4)` is `1`. The output is compatible with `md5crypt` on BSD and Linux systems.

The maximum password length for `crypt_bsdmd5` is 255 characters.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`

名前

crypt_sha256 – password hashing module using SHA–256 message hash algorithm

形式

`/usr/lib/security/$ISA/crypt_sha256.so`

機能説明

The `crypt_sha256` module is a one-way password hashing module for use with `crypt(3C)` that uses the SHA–256 message hash algorithm. The algorithm identifier for `crypt.conf(4)` and `policy.conf(4)` is `5`.

This module is designed to make it difficult to crack passwords that use brute force attacks based on high speed SHA–256 implementations that use code inlining, unrolled loops, and table lookup.

The maximum password length for `crypt_sha256` is 255 characters.

The following options can be passed to the module by means of `crypt.conf(4)`:

`rounds=`<*positive_number*>
Specifies the number of rounds of SHA-256 to use in generation of the salt; the default number of rounds is 5000. Negative values have no effect and are ignored. The minimum number of rounds cannot be below 1000.

The number of additional rounds is stored in the salt string returned by `crypt_gensalt(3C)`. For example:

```
$5,rounds=6000$nlxmTTpz$
```

When `crypt_gensalt(3C)` is being used to generate a new salt, if the number of additional rounds configured in `crypt.conf(4)` is greater than that in the old salt, the value from `crypt.conf(4)` is used instead. This allows for migration to stronger (but more time-consuming) salts on password change.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`

名前

crypt_sha512 – password hashing module using SHA–512 message hash algorithm

形式

`/usr/lib/security/$ISA/crypt_sha512.so`

機能説明

The `crypt_sha512` module is a one-way password hashing module for use with `crypt(3C)` that uses the SHA–512 message hash algorithm. The algorithm identifier for `crypt.conf(4)` and `policy.conf(4)` is 6.

This module is designed to make it difficult to crack passwords that use brute force attacks based on high speed SHA–512 implementations that use code inlining, unrolled loops, and table lookup.

The maximum password length for `crypt_sha512` is 255 characters.

The following options can be passed to the module by means of `crypt.conf(4)`:

`rounds=`<*positive_number*>
: Specifies the number of rounds of SHA-512 to use in generation of the salt; the default number of rounds is 5000. Negative values have no effect and are ignored. The minimum number of rounds cannot be below 1000.

  The number of additional rounds is stored in the salt string returned by `crypt_gensalt(3C)`. For example:

  ```
  $6,rounds=6000$nlxmTTpz$
  ```

  When `crypt_gensalt(3C)` is being used to generate a new salt, if the number of additional rounds configured in `crypt.conf(4)` is greater than that in the old salt, the value from `crypt.conf(4)` is used instead. This allows for migration to stronger (but more time-consuming) salts on password change.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`

名前

crypt_sunmd5 – password hashing module using MD5 message hash algorithm

形式

`/usr/lib/security/$ISA/crypt_sunmd5.so`

機能説明

The `crypt_sunmd5` module is a one-way password hashing module for use with `crypt(3C)` that uses the MD5 message hash algorithm. The algorithm identifier for `crypt.conf(4)` and `policy.conf(4)` is `md5`.

This module is designed to make it difficult to crack passwords that use brute force attacks based on high speed MD5 implementations that use code inlining, unrolled loops, and table lookup.

The maximum password length for `crypt_sunmd5` is 255 characters.

The following options can be passed to the module by means of `crypt.conf(4)`:

`rounds=`<*positive_number*>

Specifies the number of additional rounds of MD5 to use in generation of the salt; the default number of rounds is 4096. Negative values have no effect and are ignored, that is, the number of rounds cannot be lowered below 4096.

The number of additional rounds is stored in the salt string returned by `crypt_gensalt(3C)`. For example:

```
$md5,rounds=1000$nlxmTTpz$
```

When `crypt_gensalt(3C)` is being used to generate a new salt, if the number of additional rounds configured in `crypt.conf(4)` is greater than that in the old salt, the value from `crypt.conf(4)` is used instead. This allows for migration to stronger (but more time-consuming) salts on password change.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`

名前

crypt_unix – traditional UNIX crypt algorithm

機能説明

The `crypt_unix` algorithm is the traditional UNIX crypt algorithm. It is not considered sufficiently secure for current systems and is provided for backwards compatibility. The `crypt_sunmd5(5)`, `crypt_bsdmd5(5)`, or `crypt_bsdbf(5)` algorithm should be used instead.

The algorithm identifier for `policy.conf(4)` is `__unix__`. There is no entry in `crypt.conf(4)` for this algorithm.

The `crypt_unix` algorithm is internal to `libc` and provides the string encoding function used by `crypt(3C)` when the first character of the salt is not a "$".

This algorithm is based on a one-way encryption algorithm with variations intended (among other things) to frustrate use of hardware implementations of a key search. Only the first eight characters of the key passed to `crypt()` are used with this algorithm; the rest are silently ignored. The salt is a two-character string chosen from the set [a-zA-Z0-9./]. This string is used to perturb the hashing algorithm in one of 4096 different ways.

The maximum password length for `crypt_unix` is 8 characters.

使用法

The return value of the `crypt_unix` algorithm might not be portable among standard-conforming systems. See `standards(5)`.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	Safe

関連項目

`passwd(1)`, `crypt(3C)`, `crypt_genhash_impl(3C)`, `crypt_gensalt(3C)`, `crypt_gensalt_impl(3C)`, `getpassphrase(3C)`, `crypt.conf(4)`, `passwd(4)`, `policy.conf(4)`, `attributes(5)`, `crypt_bsdbf(5)`, `crypt_bsdmd5(5)`, `crypt_sunmd5(5)`, `standards(5)`

名前

datasets – Oracle Solaris ZFS dataset organization

機能説明

Oracle Solaris is installed to a hierarchical set of ZFS datasets with well-defined names. The ZFS datasets, composed of both file systems and volumes, are stored on a bootable ZFS storage pool in the global zone. Snapshots are also ZFS datasets, but will not be discussed here.

The datasets are created automatically by one of the installation tools: the automated installer, the text installer, or the GUI installer. For non-global zones, the datasets are created by the zone brand during the zone installation and might differ from the hierarchy described here, which applies to the `solaris(5)` brand and the global zone.

rpool Dataset

`rpool` is the default name of the top-level dataset in the bootable ZFS storage pool and is mounted at `/rpool`. Bootable ZFS storage pools have certain restrictions over normal ZFS storage pools. These restrictions are discussed in the "Oracle Solaris Administration: ZFS File Systems" section of the Oracle Solaris 11 Information Library.

Administrators can create datasets anywhere beneath the top-level dataset, but those created within the '`ROOT/be-<name>`' dataset are treated specially by `beadm(1M)` and are discussed later.

ROOT Dataset

This is a container dataset which is not mounted by Solaris during operation. It collects sets of related datasets, each set forming a ZFS Boot Environment (BE). A BE is created on installation. Additional BEs are created by `pkg`(1) and can be modified, created, or destroyed by `beadm`.

In the global zone, the ROOT dataset must be stored within a bootable ZFS storage pool, called '`rpool`' by default. For non-global zones, the ROOT dataset is stored in a dataset with the leaf name '`rpool`' beneath the zonepath of each zone.

The list below shows the datasets that are created beneath the ROOT dataset. This list uses the default storage pool name.

`rpool/ROOT/solaris`
: This is the default dataset that is created for the first boot environment installed on the system. It contains the root file system and is mounted on '/'.

  The name of the leaf component of this dataset corresponds to the name of the BE that is associated with it. Any datasets created by the administrator as children of the '`ROOT/<be-name>`' dataset will be associated with that BE, and will be created, cloned, destroyed and have snapshots taken of them as part of the BE by `beadm`. New datasets created beneath '`ROOT/<be-name>`' should have the zfs property '`canmount=noauto`' set so that the datasets are mounted by `beadm` when appropriate.

  All packaged content that is part of the image in a boot environment must fully reside within the datasets belonging to the boot environment.

`rpool/ROOT/solaris/var`
: This is a dataset associated with each boot environment that contains the `/var` file system for the BE.

Dump Dataset

`rpool/dump` is a ZFS volume created during installation and is used as the default dump device for the system crash dump facility. This dataset is shared across all BEs. The crash dump facility can be managed with `dumpadm`(1M) and the dump device can be resized using `zfs`(1M). The `rpool/dump` dataset is not present on non-global zones. Other devices can be used instead of `rpool/dump`.

Swap Dataset

`rpool/swap` is a ZFS zvol created during installation and is used as the default swap area for the system. This dataset is shared across all BEs. Use `swap`(1M) to add or remove swap volumes. Use `zfs` to resize swap volumes. The `rpool/swap` dataset is not present on non-global zones. Other swap volumes can be used instead, as well as `rpool/swap`.

`rpool/VARSHARE`

`rpool/VARSHARE` is a dataset created on installation or upgrade, and is not part of any one BE. Instead, it is shared across all BEs and is mounted automatically by that BE during boot. It contains unpackaged content from `/var`. For example, audit information is stored in `/var/audit`, core files are stored in `/var/cores`, saved crash dumps are stored in `/var/crash`, and system mail is stored in `/var/mail`.

`rpool/export/home`

`rpool/export/home` is a dataset created on installation which can be used for user home directories. It is the default location for user home directories if the -d option is not passed to `useradd`(1M) to specify a different directory. If the user home directory does not already exist and the parent directory is the mount point of a ZFS dataset (for example, `rpool/export/home`), then a child of that dataset will be created and mounted at the specified location. These datasets are shared across all BEs.

関連項目

`beadm`(1M), `dumpadm`(1M), `swap`(1M), `useradd`(1M), `zfs`(1M), `zoneadm`(1M), `zpool`(1M), `brands`(5), `filesystem`(5), `pkg`(5), `solaris`(5)

名前

device_clean – device clean programs

機能説明

Each allocatable device has a device clean program associated with it. Device clean programs are invoked by `deallocate(1)` to clean device states, registers, and any residual information in the device before the device is allocated to a user. Such cleaning is required by the object reuse policy.

Use `list_devices(1)` to obtain the names and types of allocatable devices as well as the cleaning program and the authorizations that are associated with each device.

On a system configured with Trusted Extensions, device clean programs are also invoked by `allocate(1)`, in which case the program can optionally mount appropriate media for the caller.

The following device clean programs reside in `/etc/security/lib`.

`audio_clean`	audio devices
`st_clean`	tape devices
`sr_clean`	CD-ROM devices

On a system configured with Trusted Extensions, the following additional cleaning programs and wrappers are available.

`disk_clean`	CD-ROM and other removable media devices. This program mounts the device during the execution of `allocate`, if required.
`audio_clean_wrapper`	wrapper to make audio_clean work with CDE
`wdwwrapper`	wrapper to make other cleaning programs work with CDE
`wdwmsg`	CDE dialog boxes for cleaning programs

Administrators can create device clean programs for their sites. These programs must adhere to the syntax described below.

`/etc/security/lib/`*`device-clean-program`* `[–i | –f | –s | –I] \`
`–m` *`mode`* `–u` *`user-name`* `–z` *`zone-name`* `–p` *`zone-path`* *`device-name`*

where:

`device-name`	The name of the device that is to be cleaned. Use `list_devices` to obtain the list of allocatable devices.
`-i`	Invoke boot-time initialization.
`-f`	Force cleanup by the administrator.
`-s`	Invoke standard cleanup by the user.
`-I`	Same as `-i`, with no error or warning.

The following options are supported only when the system is configured with Trusted Extensions.

`-m` *mode*	Specify the mode in which the clean program is invoked. Valid values are allo- cate and deallocate. The default mode is allocate.
`-u` *user-name*	Specify the name of user who executes the device clean program. The default user is the caller.
`-z` *zone-name*	Specify the name of the zone in which the device is to be allocated or deallocated. The default zone is the global zone.
`-p` *zone-path*	Establish the root path of the zone that is specified by *zone-name*. Default is "/".

終了ステータス

The following exit values are returned:

`0`
Successful completion.

`1`
An error. Caller can place device in error state.

`2`
A system error. Caller can place device in error state.

On a system configured with Trusted Extensions, the following additional exit values are returned:

`3`
Mounting of device failed. Caller shall not place device in error state.

`4`
Mounting of device succeeded.

ファイル

`/etc/security/lib/*` device clean programs

属性

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

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

The Invocation is Uncommitted. The Output is Not-an-interface.

関連項目

`allocate(1)`, `deallocate(1)`, `list_devices(1)`, `attributes(5)`

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

名前

dhcp – 動的ホスト構成プロトコル

機能説明

動的ホスト構成プロトコル (DHCP) を使用すると、ブート時に TCP/IP ネットワーク上のホストシステムをネットワークに合わせて自動的に構成できます。DHCP では、クライアントサーバーメカニズムが使用されます。サーバーはクライアントの構成情報を格納し、クライアントのリクエストに応じてその構成情報を提供します。この情報には、クライアントの IP アドレスと、クライアントが使用可能なネットワークサービスに関する情報を含めることができます。

このマニュアルページでは、Oracle Solaris の DHCP 実装についてのサマリーを説明します。

Oracle Solaris DHCP クライアント

Oracle Solaris DHCP クライアントは、バックグラウンドデーモン (dhcpagent(1M)) として実装されます。

IPv4 では、/etc に少なくとも 1 つの dhcp.*interface* ファイルが存在する場合に、このデーモンがブート中に自動的に開始されます。ブート中に自動的に構成されるのは、対応する /etc/dhcp.*interface* ファイルが含まれるインタフェースのみです。

IPv6 では、in.ndpd コマンドが (IPv6 Routing Advertisement メッセージに基づいて) 発行されるときに、このデーモンが自動的に開始されます。/etc/dhcp.*interface* ファイルは必要ありませんが、IPv4 DHCP も使用されていれば、このようなファイルを使用してインタフェースを「プライマリ」として指定できます。

dhcpinfo(1) コマンドを使用することによって、ブート中のシステム構成で必要なネットワークパラメータが、デーモンで受信された情報から抽出されます。/etc/default/dhcpagent ファイルでチューニング可能値を変更すると、デーモンのデフォルト動作を変更できます。このデーモンは、ifconfig(1M) ユーティリティーによって制御されます。netstat(1M) および ifconfig(1M) コマンドを使用して、デーモンのステータスをチェックします。

Oracle Solaris DHCP サーバー

Oracle Solaris DHCP サーバーは、バックグラウンドデーモン (in.dhcpd(1M)) として実装されます。このデーモンでは、ネットワーク構成情報を BOOTP または DHCP クライアントに提供できます。Oracle Solaris DHCP サービスは、dhcpmgr(1M) GUI またはコマンド行ユーティリティー dhcpconfig(1M)、dhtadm(1M)、および pntadm(1M) を使用して管理できます。

Oracle Solaris DHCP サーバーは廃止されています。代わりに、ISC DHCP サーバーを使用します。usr/share/man/man5/isc-dhcp.5 を参照してください

DHCP 構成テーブル

Oracle Solaris DHCP サーバーでは、次の 2 種類のテーブルにクライアント構成情報が格納されます:

dhcptab テーブル

構成情報のパッケージを作成して、各 DHCP クライアントに送信するために使用されるマクロおよびオプション (シンボルとも呼ばれる) が含まれます。DHCP サービスごとに、1 つの dhcptab のみが

存在します。dhtadm(1M) コマンドまたは dhcpmgr(1M) グラフィカルユーティリティーを使用すると、dhcptab(4) を表示および変更できます。dhcptab レコードの構文の詳細については、dhcptab(4) を参照してください。DHCP のオプションおよびシンボルの詳細については、dhcp_inittab(4) を参照してください。

DHCP ネットワークテーブル　DHCP ネットワークテーブルには、クライアント ID と IP アドレスおよびこれらのアドレスに関連付けられたパラメータのマッピングが含まれています。ネットワークテーブルは、名前にネットワークの IP アドレスが含まれ、pntadm コマンドまたは dhcpmgr グラフィカルユーティリティーを使用して編集、表示、および変更できます。ネットワークテーブルの詳細については、dhcp_network(4) を参照してください。

関連項目

dhcpinfo(1)、dhcpagent(1M)、dhcpconfig(1M)、dhcpmgr(1M)、dhtadm(1M)、ifconfig(1M)、in.dh

usr/share/man/man5/isc-dhcp.5

『Oracle Solaris DHCP Service Developer's Guide』

Alexander、S.、および R。Droms.RFC 2132, DHCP Options and BOOTP Vendor Extensions.Silicon Graphics, Inc. Bucknell University.1997 年 3 月

Droms, R.RFC 1534, Interoperation Between DHCP and BOOTP.Bucknell University.1993 年 10 月

Droms, R.RFC 2131, Dynamic Host Configuration Protocol.Bucknell University.1997 年 3 月

Wimer, W.RFC 1542, Clarifications and Extensions for the Bootstrap Protocol.Carnegie Mellon University.1993 年 10 月

Lemon, T. and B.Sommerfeld.RFC 4361, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4).Nominum and Sun Microsystems.February 2006.

Droms, R.RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6).Cisco Systems. July 2003.

名前

dhcp_modules – data storage modules for the DHCP service

機能説明

This man page describes the characteristics of data storage modules (public modules) for use by the Solaris Dynamic Host Configuration Protocol (DHCP) service.

Public modules are the part of the DHCP service architecture that encapsulate the details of storing DHCP service data in a data storage service. Examples of data storage services are Oracle and `ufs` file systems.

Public modules are dynamic objects which can be shipped separately from the Solaris DHCP service. Once installed, a public module is visible to the DHCP service, and can be selected for use by the service through the DHCP service management interfaces (`dhcpmgr(1M)`, `dhcpconfig(1M)`, `dhtadm(1M)`, and `pntadm(1M)`).

Public modules may be provided by Sun Microsystems, Inc or by third parties.

The Solaris DHCP service management architecture provides a mechanism for plugging in public module-specific administration functionality into the `dhcpmgr`(1M) and `dhcpconfig`(1M) utilities. This functionality is in the form of a Java Bean, which is provided by the public module vendor. This Java Bean collects public module-specific configuration from the user (you) and provides it to the Solaris DHCP service.

The Solaris DHCP service bundles three modules with the service, which are described below. There are three `dhcpsvc.conf`(4) DHCP service configuration parameters pertaining to public modules: `RESOURCE`, `PATH`, and `RESOURCE_CONFIG`. See `dhcpsvc.conf(4)` for more information about these parameters.

SUNWfiles

This module stores its data in ASCII files. Although the format is ASCII, hand-editing is discouraged. It is useful for DHCP service environments that support several hundred to a couple thousand of clients and lease times are a few hours or more.

This module's data may be shared between DHCP servers through the use of NFS.

SUNWbinfiles

This module stores its data in binary files. It is useful for DHCP service environments with many networks and many thousands of clients. This module provides an order of magnitude increase in performance and capacity over SUNWfiles.

This module's data cannot be shared between DHCP servers.

関連項目

`crontab(1)`, `dhcpconfig(1M)`, `dhcpmgr(1M)`, `dhtadm(1M)`, `pntadm(1M)`, `dhcpsvc.conf(4)`, `dhcp(5)`

『Oracle Solaris DHCP Service Developer's Guide』

名前　　environ – ユーザー環境

機能説明　　プロセスの実行を開始すると、exec 関数のファミリの中の 1 つが環境と呼ばれる文字列の配列を用意します (exec(2) を参照)。規約上、これらの文字列は *variable* (変数) = *value* (値) という形式をとります (たとえば、PATH=/sbin:/usr/sbin)。この環境変数により、プログラムはプログラムの環境に関する情報を利用できるようになります。

環境中に名前を設定するには、sh(1) における *name*=*value* 引数と export コマンドを使用するか、あるいは exec 関数のうちの 1 つを使用します。.profile ファイル内でエクスポートされる MAIL、PS1、PS2、IFS などのシェル変数と環境変数との間で矛盾を生じさせるのは避けてください。詳細は profile(4) を参照してください。

アプリケーションが利用できる環境変数を以下に示します。これらの変数は、実行環境において設定されていることが期待されています。

HOME
ユーザーのログインディレクトリ名。パスワードファイル (passwd(4) を参照) に基づいて login(1) が設定します。

LANG
国際化情報を指定するのに用いる文字列。ユーザーはこの情報を利用して、さまざまな国の慣習に従って作業を進めることができます。setlocale(3C) 関数の引数 locale に "" を設定して呼び出すと、この関数は環境変数 LANG を調べます。特定のカテゴリに対する環境変数が設定されていない、あるいは NULL に設定されている場合は、LANG がデフォルト時のロケールとして使用されます。ただし LC_ALL が空文字列でない正当な値に設定されていれば、変数 LANG やほかの LC_* 変数の代わりにその値が使用されます。たとえば、setlocale(LC_CTYPE, "") のように呼び出されると、setlocale() は、まず環境変数 LC_CTYPE が設定されていて NULL でないことを問い合わせます。LC_CTYPE が未設定または NULL の場合は、setlocale() は次に環境変数 LANG を調べて、それが設定されていて NULL でないことを確認します。LANG と LC_CTYPE の両方が未設定または NULL の場合、デフォルトの C ロケールを使用して LC_CTYPE カテゴリを設定します。

ほとんどのコマンドは、ほかの処理を行う前に setlocale(LC_ALL, "") を呼び出します。この関数を用いて適切な環境変数を設定することにより、コマンドをさまざまな国の慣習に従って使用できるようになります。

以下に示す環境変数が各カテゴリの setlocale(3C) と対応しています。

LC_ALL
空文字列でない正当な値に設定されていれば、LANG およびすべての LC_* 変数の値に代わって使用されます。

LC_COLLATE
使用する文字照合順序を指定します。このカテゴリにあてはまる情報は、localedef(1) コマンドによって作成されるデータベース内に格納されています。この環境変数は strcoll(3C) および strxfrm(3C) に影響を及ぼします。

LC_CTYPE
文字分類、文字変換および複数バイト文字の幅を指定します。LC_CTYPE が正当な値に設定されていると、呼び出し側のユーティリティーは、該当するロケールに合った文字を含んでいるテキストやファイル名を表示したり処理したりできます。例: 各文字の幅が 1 から 3 バイトの拡張 UNIX コード (EUC) 文字、およびカラム幅が 1 から 3 の EUC 文字。デフォルトの C ロケールは、7 ビットの ASCII 文字セットに対応していて、ISO 8859-1 の文字だけが有効です。このカテゴリにあてはまる情報は、localedef() コマンドによって作成されるデータベース内に格納されています。ctype(3C)、mblen(3C)、および他の多くのコマンド (たとえば、cat(1)、ed(1)、ls(1)、vi(1)) がこの環境変数を使用します。

LC_MESSAGES
使用するメッセージデータベースの言語を指定します。たとえば、アプリケーションは、フランス語用のメッセージデータベースとドイツ語用のメッセージデータベースを使い分けることができます。メッセージデータベースは、mkmsgs(1) コマンドで作成します。exstr(1)、gettxt(1)、srchtxt(1)、gettxt(3C)、gettext(3C) がこの環境変数を使用します。

LC_MONETARY
特定のロケールで使用する通貨記号および区切り記号を指定します。このカテゴリにあてはまる情報は、localedef(1) コマンドによって作成されるデータベース内に格納されています。localeconv(3C) がこの環境変数を使用します。

LC_NUMERIC
小数点および千の位の区切り記号を指定します。このカテゴリにあてはまる情報は、localedef() コマンドによって作成されるデータベース内に格納されています。デフォルト値の C ロケールは、小数点にピリオド (.) を指定し、千の位の区切り記号は指定しません。localeconv(3C)、printf(3C)、strtod(3C) がこの環境変数を使用します。

LC_TIME
日付と時間の書式を指定します。このカテゴリにあてはまる情報は、localedef() で指定されるデータベース内に格納されています。デフォルト値の C ロケールは、アメリカ合衆国の日付と時間の書式に対応しています。多くのコマンドおよび関数がこの環境変数を使用します。例: at(1)、calendar(1)、date(1)、strftime(3C)、getdate(3C) などです。

MSGVERB
stderr にメッセージを表示するとき、fmtmsg がどの標準書式メッセージコンポーネントを使用するかを制御します (fmtmsg(1) および fmtmsg(3C) を参照)。

NETPATH
コロンで区切られたネットワーク識別子のリストです。ネットワーク識別子とは、システムの「ネットワーク選択」コンポーネントが、アプリケーション固有のデフォルトのネットワーク検索パスを提供するために 使用する文字列で

す。ネットワーク識別子は、NULL 以外の文字からなり、長さは 1 以上である必要があります。最大長は制限されていません。通常は、システム管理者がネットワーク識別子を選択します。ネットワーク識別子は、/etc/netconfig ファイルのすべての記述項の最初のフィールドでもあります。したがって NETPATH は、/etc/netconfig ファイルへのリンクおよびそのファイルのネットワーク記述項に入っているネットワークに関する情報を提供します。/etc/netconfig は、システム管理者によって維持されます。getnetpath(3NSL) で説明したライブラリルーチンが、NETPATH 環境変数にアクセスします。

NLSPATH
メッセージカタログを見つけるために catopen(3C) と gettext(3C) が使用する、一連のテンプレートを指定します。各テンプレートは、オプションの接頭辞、1 つまたは複数の置換フィールド、ファイル名、およびオプションの接尾辞からなります。例:

```
NLSPATH="/system/nlslib/%N.cat"
```

この例では、catopen() がすべてのメッセージカタログをディレクトリ /system/nlslib から見つける必要があることを定義します。このディレクトリのカタログ名は、catopen( ) に指定された *name* パラメータの %N に接尾辞の .cat がついています。

置換フィールドは、後ろに単一文字のキーワードがついた % 記号からなります。現在定義されているキーワードを以下に示します。

%N
catopen() に指定された *name* パラメータの値

%L
LANG または LC_MESSAGES の値

%l
LANG または LC_MESSAGES の言語要素

%t
LANG または LC_MESSAGES の地域要素

%c
LANG または LC_MESSAGES のコードセット要素

%%
単一の % 文字

現在定義されていない値を指定した場合は、空の文字列に置換されます。分離文字の下線文字 (_) およびピリオド (.) は、%t および %c の各置換フィールドには入っていません。

NLSPATH で定義されるテンプレートは、コロン (:) で区切られます。先頭のコロンまたは隣接する 2 つのコロン (::) は、%N と同等の意味です。例:

```
NLSPATH=":%N.cat:/nlslib/%L/%N.cat"
```

この例では、要求されたメッセージカタログを *name*、*name*.cat、および /nlslib/$LANG/*name*.cat から見つけるよう catopen() に指示しています。gettext() の場合、%N には自動的に messages にマップされます。

NLSPATH が設定されていないか NULL に設定されている場合、catopen() と gettext() は setlocale(3C) を呼び出します。setlocale3C は LANG と LC_* 変数を調べてメッセージカタログの場所を探します。

Solaris における拡張された %L の解釈には、gettext(1)、gettext(3C)、catopen(3C)、setlocale(3C)、および locale_alias(5) で説明されているように、受け入れられたロケール名別名のサポートが含まれます。

通常、NLSPATH はシステム全体 (/etc/profile 中) に設定されるので、プログラムもユーザーもメッセージカタログに関連付けられた場所や命名規約を意識する必要はありません。

PATH
: sh(1)、time(1)、nice(1)、nohup(1) などのユーティリティが、不完全なパス名しかわからないファイルを探すときに使用する一連のディレクトリ接頭辞です。複数の接頭辞はコロン (:) で区切られます。login(1) は PATH=/usr/bin を設定します。詳細は、sh(1) を参照してください。

SEV_LEVEL
: 重大度レベルを定義し、文字列をその重大度レベルと対応させ、標準書式エラーメッセージで出力します。addseverity(3C)、fmtmsg(1)、および fmtmsg(3C) を参照してください。

TERM
: 出力を行う端末の種類です。この情報は vi(1) などのコマンドによって使用され、その端末の特殊な機能が利用できます。

TZ
: タイムゾーンの情報です。この環境変数の内容は、ctime(3C)、localtime(3C)、strftime(3C)、mktime(3C) の各関数がデフォルトのタイムゾーンを変更するのに使用します。TZ の値の形式は、次の 2 つのうちのどちらかです (見やすくするために、スペースを挿入しています)。

  ```
  :characters
  ```

  または

  ```
  std offset dst offset, rule
  ```

  TZ が最初の形式である場合 (つまり、最初の文字がコロン (:) である場合)、あるいは、TZ が 2 番目の形式でない場合、TZ は、(もしあれば) 先頭のコロンを無視して、/usr/share/lib/zoneinfo/ に関連するタイムゾーンデータベースファイルへのパスを示します。

そうでない場合、つまり、TZ が 2 番目の形式である場合、TZ は次のように展開されます。

stdoffset[*dst*[*offset*][,*start*[/*time*],*end*[/*time*]]]

std と *dst*
標準のタイムゾーン (*std*) または代替のタイムゾーン (*dst*、サマータイムなど) の名前を指定します。名前の長さは 3 バイト以上 {TZNAME_MAX} バイト以下です。*std* だけが必須です。*dst* を省略した場合、このタイムゾーンに代替の時間は表示されません。これらのフィールドは、それぞれ次の 2 つの形式、つまり引用付きと引用なしのどちらかで指定できます。

- 引用付き形式の場合、最初の文字は小なり記号 (<) で、最後の文字が大なり記号 (>) です。これら引用記号の間にある文字はすべて、現在のロケールにおける移植可能な文字セットの英数字文字、プラス記号 (+)、またはマイナス記号 (-) です。この場合、*std* と *dst* フィールドは引用記号を含みません。
- 引用なし形式の場合、std と dst フィールド内のすべての文字は、現在のロケールにおける移植可能な文字セットの英数字文字です。

どちらかのフィールドが 3 バイトより小さい場合 (*dst* を省略した場合を除く)、{TZNAME_MAX} バイトより大きい場合、あるいは、前述の文字以外の文字を使用した場合、これらのフィールドの解釈方法は定義されていません。

offset
協定世界時に合わせるために、ローカル時間に追加すべき値を示します。offset の形式は以下のとおりです。

hh[:*mm*[:*ss*]]

分 (*mm*) および秒 (*ss*) はオプションです。時間 (*hh*) は必須で、単一の数字で指定することもできます。*std* の後では必ず *offset* を指定しなければなりません。*dst* の後に *offset* を指定しないと、サマータイムは標準時間の 1 時間先と見なされます。1 つまたは複数の数字による指定が可能です。この値はつねに 10 進数として解釈されます。時間は 0 と 24 の間の値で、分 (および秒) を指定する場合は 0 と 59 の間の値で指定する必要があります。この範囲外の値を指定した場合の動作は予測できません。- を値の前に付けると、タイムゾーンはグリニッジ子午線の東です。それ以外の場合は、グリニッジ子午線の西になります (オプションの + 記号で示すこともできます)。

start/*time*, *end*/*time*
サマータイムに切り替える日時と、サマータイムから元に戻る日時を示します。*start*/*time* は、標準時間からサマータイムに切り替える日時を示し、*end*/*time* は、標準時間に戻る日時を示します。各 *time* フィールドは、変更を行う時間 (現在のローカル時間) を示します。

start および *end* は、以下のいずれかの形式をとります。

`J`*n*
ユリウス日 *n* (1 ≤ *n* ≤ 365)。うるう年は考えません。つまり毎年 2 月 28 日は 59 で、3 月 1 日は 60 となります。この書式では 2 月 29 日を扱えません。

n
0 から始まるユリウス日 *n* (1 ≤ n ≤ 365)。2 月 29 日を扱えるように、うるう年を考慮した書式です。

`M`*m.n.d*
m 月 (1 ≤ *m* ≤ 12) *n* 週 (1 ≤ *n* ≤ 5) の *d* 番目の曜日 (0 ≤ *d* ≤ 6)。週 5 は、「*m* 月の最後の *d* 番目の曜日」になります (最後の d 番目の曜日は、月の第 4 週または第 5 週に来ます)。週 1 は、その月の最初の *d* 番目の曜日が来る週を意味します。曜日ゼロは日曜日です。

これらのオプションのフィールドがない場合は、システム固有のデフォルト値を使用して *start* および *end* を指定します。

time は *offset* と同じ書式を用います。ただし、time では先行符号 (– または +) を付けることはできません。*time* を指定しない場合、デフォルト値は 02:00:00 です。

関連項目

`cat(1)`, `date(1)`, `ed(1)`, `fmtmsg(1)`, `localedef(1)`, `login(1)`, `ls(1)`, `mkmsgs(1)`, `nice(1)`, `nohup(1)`, `sh(1)`, `sort(1)`, `time(1)`, `vi(1)`, `exec(2)`, `addseverity(3C)`, `catopen(3C)`, `ctime(3C)`, `ctype(3C)`, `fmtmsg(3C)`, `getdate(3C)`, `getnetpath(3NSL)`, `gettext(3C)`, `gettxt(3C)`, `localeconv(3C)`, `mblen(3C)`, `mktime(3C)`, `printf(3C)`, `setlocale(3C)`, `strcoll(3C)`, `strftime(3C)`, `strtod(3C)`, `strxfrm(3C)`, `TIMEZONE(4)`, `netconfig(4)`, `passwd(4)`, `profile(4)`, `locale_alias(5)`

名前

eqnchar – special character definitions for eqn

形式

```
eqn  /usr/share/lib/pub/eqnchar filename |  troff options
neqn  /usr/share/lib/pub/eqnchar filename |  troff options
```

機能説明

The eqnchar command contains nroff(1) and troff(1) character definitions for constructing characters that are not available on the Graphic Systems typesetter. These definitions are primarily intended for use with eqn(1) and neqn(1). It contains definitions for the characters listed in the following table.

ciplus	⊕	| |	‖	*square*	□
citimes	⊗	*langle*	⟨	*circle*	○
wig	~	*rangle*	⟩	*blot*	□
-wig	≃	*hbar*	ℏ	*bullet*	•
>wig	≳	*ppd*	⊥	*prop*	∝
<wig	≲	*<->*	↔	*empty*	∅
=wig	≅	*<=>*	⇔	*member*	∈
star	⋆	|<	⊀	*nomem*	∉
bigstar	✳	|>	⊁	*cup*	∪
=dot	≐	*ang*	∠	*cap*	∩
orsign	∨	*rang*	⌡	*incl*	⊑
andsign	∧	*3dot*	⋮	*subset*	⊂
=del	≙	*thf*	∴	*supset*	⊃
oppA	∀	*quarter*	¼	*!subset*	⊆
oppE	∃|	*3quarter*	¾	*!supset*	⊇
angstrom	Å	*degree*	°		

ファイル

/usr/share/lib/pub/eqnchar

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

関連項目

eqn(1), nroff(1), troff(1), attributes(5)

名前

extendedFILE – enable extended FILE facility usage

形式

```
$ ulimit -n N_file_descriptors
$ LD_PRELOAD_32=/usr/lib/extendedFILE.so.1 application [arg...]
```

機能説明

The `extendedFILE.so.1` is not a library but an enabler of the extended FILE facility.

The extended FILE facility allows 32-bit processes to use any valid file descriptor with the standard I/O (see `stdio(3C)`) C library functions. Historically, 32-bit applications have been limited to using the first 256 numerical file descriptors for use with standard I/O streams. By using the extended FILE facility this limitation is lifted. Any valid file descriptor can be used with standard I/O. See the NOTES section of `enable_extended_FILE_stdio(3C)`.

The extended FILE facility is enabled from the shell level before an application is launched. The file descriptor limit must also be raised. The syntax for raising the file descriptor limit is

```
$ ulimit -n max_file_descriptors
$ LD_PRELOAD_32=/usr/lib/extendedFILE.so.1 application [arg...]
```

where *max_file_descriptors* is the maximum number of file descriptors desired. See `limit(1)`. The maximum value is the same as the maximum value for `open(2)`.

環境変数

The following environment variables control the behavior of the extended FILE facility.

`_STDIO_BADFD` — This variable takes an integer representing the lowest file descriptor, which will be made unallocatable. This action provides a protection mechanism so that applications that abuse interfaces do not experience silent data corruption. The value must be between 3 and 255 inclusive.

`_STDIO_BADFD_SIGNAL` — This variable takes an integer or string representing any valid signal. See `signal.h(3HEAD)` for valid values or strings. This environment variable causes the specified signal to be sent to the application if certain exceptional cases are detected during the use of this facility. The default signal is `SIGABRT`.

使用例

例 1 Limit the number of file descriptors and FILE standard I/O structures.

The following example limits the number of file descriptors and FILE standard I/O structures to 1000.

```
$ ulimit -n 1000
$ LD_PRELOAD_32=/usr/lib/extendedFILE.so.1 application [arg...]
```

例 2 Enable the extended FILE facility.

The following example enables the extended FILE facility. See `enable_extended_FILE_stdio(3C)` for more examples.

例 2 Enable the extended FILE facility. (続き)

```
$ ulimit -n 1000
$ _STDIO_BADFD=100 _STDIO_BADFD_SIGNAL=SIGABRT \
   LD_PRELOAD_32=/usr/lib/extendedFILE.so.1  \
   application [arg ...]
```

例 3 Set up the extended FILE environment and start the application.

The following shell script first sets up the proper extended FILE environment and then starts the application:

```
#!/bin/sh
if [ $# = 0 ]; then
   echo "usage: $0 application [arguments...]"
   exit 1
fi
ulimit -n 1000
# _STDIO_BADFD=196; export _STDIO_BADFD
# _STDIO_BADFD_SIGNAL=SIGABRT; export _STDIO_BADFD_SIGNAL
LD_PRELOAD_32=/usr/lib/extendedFILE.so.1; export LD_PRELOAD_32
"$@"
```

ファイル

`/usr/lib/extendedFILE.so.1` enabling library

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library (32–bit)
Interface Stability	Committed
MT-Level	Safe

関連項目

`limit(1)`, `open(2)`, `enable_extended_FILE_stdio(3C)`, `fdopen(3C)`, `fopen(3C)`, `popen(3C)`, `signal.h(3HEAD)`, `stdio(3C)`, `attributes(5)`

警告

The following displayed message

```
Application violated extended FILE safety mechanism.
Please read the man page for extendedFILE.
Aborting
```

is an indication that your application is modifying the internal file descriptor field of the `FILE` structure from standard I/O. Continued use of this extended FILE facility could harm your data. Do not use the extended FILE facility with your application.

名前

extensions – localedef extensions description file

機能説明

A localedef extensions description file or *extensions* file defines various extensions for the `localedef(1)` command.

The localedef extensions description file provides:

- EUC code set width information via the `cswidth` keyword: `cswidth bc1 : sw1,bc2 : sw2,bc3 : sw3`where `bc1`, `bc2`, and `bc3` indicate the number of bytes (byte count) per character for EUC codesets `1`, `2`, and `3`, respectively. `sw1`, `sw2`, and `sw3` indicate screen width for EUC codesets `1`, `2`, and `3`, respectively.
- Other extensions which will be documented in a future release.

関連項目

`locale(1)`, `localedef(1)`, `environ(5)`, `locale(5)`

名前

fedfs – overview of Federated Filesystem (FedFS)

機能説明

The NFSv4 Working Group of the Internet Engineering Task Force (IETF) has been working to standarize a set of protocols which together permit construction and maintenance of a federated filesystem, where many different file servers together share a single namespace. These protocols make it possible to create a multi-vendor global namespace.

Solaris currently has support for two of the three FedFS protocols:

DNS SRV records
: The Solaris automounter (automount(1M)) supports the `-fedfs` map, which by default will mount `/nfs4/`*dns_domain* by looking up a DNS SRV record for the domain. This map is enabled by default in `/etc/auto_master`.

LDAP-based referrals
: The Solaris NFS server will follow FedFS referrals by looking up location data in LDAP, guided by the schema specified by the FedFS specs.

使用例

例 1 Using the `-fedfs` Automounter Map

To mount a path like `/nfs4/cthon.org`, your NFS client should be set up to use DNS such that the domain's DNS server should answer a query for the `_nfs-domainroot._tcp` SRV record like this:

```
$ nslookup '-q=srv' _nfs-domainroot._tcp.cthon.org
Server:         1.2.3.4
Address:        1.2.3.4#53

_nfs-domainroot._tcp.cthon.org  service = 0 0 2049 root-a.cthon.org.
_nfs-domainroot._tcp.cthon.org  service = 1 1 2049 root-n.cthon.org.

$ cd /nfs4/cthon.org

$ df .
/nfs4/cthon.org    (root-a.cthon.org,root-b.cthon.org:/.domainroot/\
cthon.org):120379963 blocks 120379963 files
```

例 2 Using `nsdbparams` to Set Up Communications with an NSDB

The following example sets up communications with an NSDB called `nsdb.cthon.org` and makes it the default NSDB:

```
# nsdbparams update -D cn=Manager,dc=cthon,dc=org -w cthon.org \
nsdb.cthon.org
# nsdbparams show nikon.us.example.com
nikon.us.example.com:389
     default bind DN: cn=Manager,dc=cthon,dc=org
     default bind PW: cthon.org
     default NCE: dc=cthon,dc=org
     sectype: FEDFS_SEC_NONE
```

例 2 Using nsdbparams to Set Up Communications with an NSDB *(続き)*

```
# nsdbparams set nsdb.cthon.org
# nsdbparams get
default nsdb: nsdb.cthon.org
default port: 389
```

例 3 Using the NSDB Tools

The following commands illustrate the use of the NSDB tools available in Oracle Solaris.

```
# nsdb-nces
Host: nsdb.cthon.org:389
  namingContext 'dc=cthon,dc=org' is a FedFS NCE, DIT starts at ''

# nsdb-list
SDB: nsdb.cthon.org:389, dc=cthon,dc=org
  FSN UUID: 7cc0bf04-5459-11e1-8083-80093d11d889
     FSL UUID: 7cc33c02-5459-11e1-8084-00093d11d889 = filer-a:/tmp
  FSN UUID: db48f160-5858-11e1-b459-80093d11d889
     FSL UUID: db4998c2-5858-11e1-b45a-00093d11d889 = filer-j:/tmp

# nsdb-resolve-fsn 7cc0bf04-5459-11e1-8083-80093d11d889
For FSN UUID 7cc0bf04-5459-11e1-8083-80093d11d889
  FSL UUID: 7cc33c02-5459-11e1-8084-00093d11d889
    Location: filer-a:/tmp
```

例 4 Using nfsref to Create a Referral

The following sequence of commands illustrates the use of nfsref to create a referral.

```
# nsdb-list
NSDB: nsdb.cthon.org:389, dc=cthon,dc=org
  FSN UUID: 7cc0bf04-5459-11e1-8083-80093d11d889
    FSL UUID: 7cc33c02-5459-11e1-8084-00093d11d889 = filer-a:/tmp
  FSN UUID: db48f160-5858-11e1-b459-80093d11d889
    FSL UUID: db4998c2-5858-11e1-b45a-00093d11d889 = filer-j:/tmp

# nfsref -t nfs-fedfs add /root/tools filer-k:/tools filer-x:/tools
Enter password for cn=Manager,dc=cthon,dc=org:
Created reparse point /root/tools

# nsdb-list
NSDB: nsdb.cthon.org:389, dc=cthon,dc=org
  FSN UUID: 7cc0bf04-5459-11e1-8083-80093d11d889
    FSL UUID: 7cc33c02-5459-11e1-8084-00093d11d889 = filer-a:/tmp
  FSN UUID: db48f160-5858-11e1-b459-80093d11d889
    FSL UUID: db4998c2-5858-11e1-b45a-00093d11d889 = filer-j:/tmp
  FSN UUID: 004b2382-9663-11e1-8c79-80093d11d888
    FSL UUID: 004da2b0-9663-11e1-8c7a-00093d11d888 = filer-k:/tools
```

例 4 Using nfsref to Create a Referral (続き)

```
FSL UUID: 004e9bac-9663-11e1-8c7b-00093d11d888 = filer-x:/tools
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

関連項目

automount(1M), nfsref(1M), nsdb-list(1M), nsdbparams(1M), attributes(5)

RFC 5716: Requirements for Federated File Systems

名前 filesystem – file system organization

形式 /

機能説明 The file system is a hierarchical structure of descendent file systems and directories that are used to organize system-related components and binaries as well as non-system-related components and home directories in the Oracle Solaris OS. By default, the root file system is installed within a ZFS root pool and specifically, is a ZFS file system with separate directories of system-related components, such as `etc`, `usr`, and `var`, that must be available for the system to function correctly. After a system is installed, the root of the Solaris file system is mounted, which means files and directories are accessible.

All subdirectories of the root file system that are part of the Oracle Solaris OS, with the exception of `/var`, must be contained in the same file system as the root file system. In addition, all packaged software must reside in the root pool, with the exception of the swap and dump devices. A default swap device and dump device are created automatically as ZFS volumes in the root pool when a system is installed.

The following file system content descriptions make use of platform, platform-dependent, platform-independent, and platform-specific terms. Platform refers to a system's Instruction Set Architecture or processor type as returned by the `uname -i` command. Platform-dependent refers to a file that is installed on all platforms and whose contents vary depending on the platform. Like a platform-dependent file, a platform-independent file is installed on all platforms. However, the contents of the latter type remains the same on all platforms. An example of a platform-dependent file is compiled, executable program. An example of a platform-independent file is a standard configuration file, such as `/etc/hosts`. Unlike a platform-dependent or a platform- independent file, the platform-specific file is installed only on a subset of supported platforms. Most platform-specific files are gathered under `/platform` and `/usr/platform`.

Root File System The root file system contains files and directories that are critical for system operation, such as the kernel, the device drivers, and the programs used to boot the system. These components are described below. The root (`/`) directory also contains mount point directories where local and remote file systems can be attached to the file system hierarchy.

`/`
: Root directory of the entire file system name space. This is a special file system that is mounted by the kernel at system boot time.

`/boot`
: Directory that contains files and executables that are needed for booting the system.

`/bin`
: Symbolic link to the `/usr/bin` directory that contains system executables and scripts.

`/dev`
: Directory that contains special device files. Typically, device files are built to match the kernel and hardware configuration of the system.

`/devices`
Mount point directory for the `devfs` file system that manages the device name space.

`/etc`
Directory that contains platform-dependent administrative and configuration files and databases that are not shared among systems. This directory defines the system's identity. An approved installation location for bundled Solaris software.

`/export/home` or `/home`
Directory or file system mount point for user home directories, which store user files. By default, the `/home` directory is an automounted file system.

`/kernel`
Directory of platform-dependent loadable kernel modules required as part of the boot process. It includes the generic part of the core kernel that is platform-independent, `/kernel/genunix`. See `kernel(1M)`. An approved installation location for bundled Oracle Solaris software and for add-on system software.

`/lib`
Directory that contains core system libraries. Historically, this directory contained essential library components for system startup.

`/media`
Directory for accessing removable media that is automatically mounted.

`/mnt`
Default temporary mount point directory for file systems. This empty directory is used to temporarily mount a file system.

`/net`
Temporary mount point directory for file systems that are mounted by the automounter.

`/opt`
Directory for unbundled application packages.

`/platform`
Directory of platform-specific objects that need to reside in the root file system. It contains a series of directories, one per supported platform. The semantics of the series of directories is equivalent to `/` (root).

`/proc`
Mount point directory for the process file system.

`/root`
Home directory for the root user.

`/rpool`
Mount point directory for the ZFS boot-related components. By default, the root pool is named `rpool` during installation.

`/sbin`
Symbolic link to the `/usr/sbin` directory.

`/system`
Mount point directory for the contract (`CTFS`) and object (`OBJFS`) file systems.

`/tmp`
Directory that contains temporary files that are removed during a boot operation.

`/usr`
Directory that contains platform-dependent and platform-independent binaries and files. The `/usr/share` subdirectory contains platform-independent files. The rest of the `/usr` directory contains platform-dependent files.

`/usr/bin`
Directory that contains platform-dependent, user-invoked executables. These are commands that users expect to be run as part of their normal `$PATH`. An approved installation location for bundled Oracle Solaris software. The analogous location for unbundled system software or for applications is `/opt/packagename/bin`.

`/usr/lib`
Directory that augments the contents of `/lib` with additional system libraries, and other supporting files that are required by programs at runtime.

`/usr/java`
Directory that contains Java files and executables.

`/usr/sbin`
Contains essential executables used in the booting process and in manual system recovery. Historically, this directory was needed to recover the system before the `/usr` file system was mounted. In this Oracle Solaris release, `/usr` is a directory, not a separate file system and is available when the root file system is mounted.

`/var`
Directory or file system that contains varying files that are unique to a system but can grow to an arbitrary or variable size. An example is a log file. An approved installation location for bundled Oracle Solaris software.

`/var/share`
Directory that is shared across several boot environments, and is resident on a separate dataset beneath the `zpool` containing boot environments. Packages should not deliver content here, since package metadata in a given boot environment may not reflect the current content of `/var/share`.

Packages may deliver directories to `/var/.migrate`, which will be created automatically beneath `/var/share` during boot. See `pkg`(5) and the IPS Developer's Guide for a description of how to share data across boot environments.

By default, the following directories are shared:

```
/var/share/audit
/var/share/cores
/var/share/crash
/var/share/mail
```

Symlinks are delivered to `/var` to point to each shared directory.

If existing datasets attempt to mount on one of those symlinks (for example, users with an existing dataset that normally mounts on `/var/mail`), then those datasets will be mounted beneath `/var/share`, since filesystem mounts traverse symlinks. The dataset will still be accessible through the original mountpoint.

`/var/tmp`
Directory that contains files that vary in size or presence during normal system operations. The content of this directory is not removed during a boot operation. It is possible to change the default behavior for `/var/tmp` to clear all of the files except editor temporary files by setting the `clean_vartmp` property value of the `rmtmpfiles` service. This is done with the following commands:

```
# svccfg -s svc:/system/rmtmpfiles setprop\
            options/clean_vartmp = "true"
# svcadm refresh svc:/system/rmtmpfiles:default
```

The `solaris.smf.value.rmtmpfiles` authorization is required to modify this property.

関連項目

isainfo(1), svcs(1), uname(1), automount(1M), automountd(1M), boot(1M), init(1M), kernel(1M), mount(1M), svcadm(1M), svccfg(1M), zfs(1M), zpool(1M), mount(2), Intro(4), proc(4), , ctfs(7FS), devfs(7FS), objfs(7FS),

名前

fmri – Fault Managed Resource Identifier

機能説明

Oracle Solaris subsystems such as the Service Management Facility (see smf(5)), the Fault Management Daemon (see fmd(1M), fmadm(1M)), and the Image Packaging System (see pkg(5)) identify each element they manage using a *fault managed resource identifier*, abbreviated as FMRI.

FMRIs are used to identify both hardware resources and software resources and abstractions. For hardware resources the FMRIs are usually an implementation detail, and when a component is diagnosed as faulty it is the FRU (field replaceable unit) location label which is used to identify it. If no FRU label is available, the FMRI is used. For software abstractions such as an SMF service or a pkg(5) software package the FMRI (often in string form) is the one and only means of naming the managed resource (abbreviations aside).

All FMRIs include an indication of the FMRI scheme adhered to, and the version of that FMRI scheme in use. Once the scheme and version are known, the remainder of the FMRI is interpreted as specified in the formal definition of that FMRI scheme version. Schemes exists for `svc`, `pkg`, `cpu`, `hc` (hardware component), `dev` (device), and a number of others.

The formal definition of a given version of a particular FMRI scheme is specified as an unordered series of FMRI members specified as (member-name, data-type, data-semantics) tuples. An instance of an FMRI is a series of (name, type, value) tuples wherein the name and type match the FMRI specification and the value provided is a value of the given type and derived and interpreted according to the semantics described in the formal FMRI definition for that scheme and version. In Oracle Solaris C library APIs such as libfmevent(3LIB), FMRIs are represented as `nvlists` using the libnvpair(3LIB) library.

It is often more convenient to represent a given FMRI instance as a string, such as in command line interaction or in command output. The specification of each FMRI scheme version includes rules for formatting an FMRI of that scheme and version as a string instead of an name-value pair list, and given an FMRI string those same rules can be applied in reverse to recover the more-formal FMRI form. In some cases the string form of an FMRI can include a reduced amount of information as compared to the formal form, but still provide enough information to identify the resource for administrative purposes. The string form of an FMRI usually does not include the FMRI version number, and so the version is understood to be that whose formal FMRI-to-string formatting rules the given FMRI string matches. The string form of an FMRI is not unlike a URI as outlined in RFC 2396 which we use as a syntactical model. In particular, we reserve the same set of characters `:`, `;`, `=`, `@`, `/`, `&`, `'`, `$` and `,` and require the same escaped encoding conventions should these characters appear in an FMRI member value that is rendered in the string.

FMRI Members

The formal definition of a given version of a particular FMRI scheme is specified as an unordered series of:

```
(member-name, data-type, data-semantics)
```

tuples, some of which are required and some of which are optional. The formal definition also specifies how to render the FMRI in string form.

`member-name`
This associates a name with the FMRI member, with characters drawn from the following alphabet:

```
A-Z
a-z
'_' (underscore)
'-' (hyphen)
'.' (period)
```

`data-type`
This specifies the data type for this member. Legal primitive datatypes are listed in the following table, and for all primitive types we also define an array of that type as a legal datatype. The primitive integer and double datatypes are defined as in C, and we add boolean value and string primitive datatypes:

```
Primitive Datatype   Description
-------------------------------------------
boolean_value        Values can be 0 or 1
string               ASCII string
[u]int8              (Un)signed 8-bit integer
[u]int16             (Un)signed 16-bit integer
[u]int32             (Un)signed 32-bit integer
[u]int64             (Un)signed 64-bit integer
double               Double precision floating point
```

For each primitive datatype we form an aggregate datatype that is an ordered array of the primitive type:

```
Primitive Datatype    Description
-----------------------------------------------
boolean_value array   Array of boolean_value
string array          Array of ASCII strings
[u]int8 array         Array of (un)signed 8-bit integer
[u]int16 array        Array of (un)signed 16-bit integer
[u]int32 array        Array of (un)signed 32-bit integer
[u]int64 array        Array of (un)signed 64-bit integer
double array          Array of double
```

A further aggregate datatype is defined - the `nvlist` (name- value list). An `nvlist` is an unordered list of (`list-member-name, list-member-datatype`) in which `list-member-name` names the member using the same alphabet as for `member-name` above, and `list-member-datatype` is any primitive or aggregate datatype including `nvlist` itself. We term the members of an `nvlist`: `nvpairs` (name-value pairs).

A special type of `nvlist` is an `fmri` which is an `nvlist` that conforms to one of the FMRI scheme versions defined below.

```
Synthetic Datatype    Description
-----------------------------------------------------------
nvlist                Unordered list of nvpairs
```

```
nvlist array          Array of nvlist
fmri                  An nvlist that specifies an FMRI
                      scheme and version and includes other
                      members so as to be a valid FMRI
                      in that scheme version.
```

`data-semantics`
The data semantics for a given member in an FMRI definition specify how the values for that member should be interpreted, that is, their meaning. The data semantics can constrain the set of legal values that the member can assume to a subset of those representable by the member data type.

Stabilities

Some FMRI schemes, such as `svc` for SMF, are administrator-facing while others are more of a private implementation detail. Correspondingly, the definition of some FMRI schemes constitute a `Committed` interface, while the definition of others form a `Private` interface. See `attributes(5)`. As with all `Private` interfaces the definition of `Private` FMRI components is subject to possibly incompatible change at any time, and you should not consume such interfaces nor rely on the presence or assumed semantics of private FMRI members.

In defining an FMRI scheme below we list a series of (`member-name, data-type, data-semantics`), specify which members are required and which are optional, and the rules for rendering an FMRI in string form. In terms of stability assignments, there are two stability levels associated with each member:

- Stability of the (`member-name, data-type`) pair
- Stability of the corresponding data-semantics

The default in both cases is `Private`. When the stability of both aspects of a member are identical (the common case) we write the stability level just once. When they differ then we write the member name and data type stability first, followed by a comma and the data semantics stability level. Today the only stability levels employed in FMRIs are `Committed`, `Private`, and `Not An Interface` but others are allowed. The semantics of a `Committed` FMRI component do not change incompatibly for given scheme and version thereof.

FMRI Scheme and Version Identification

All FMRI definitions are required to include members for the scheme name and major version thereof. The name/datatype and data semantics stabilities are `Committed`, thus all FMRI definitions are required to begin as follows:

```
Member Name      Data Type    Stability
---------------------------------------
scheme           string       Committed
version          uint8        Committed
```

`scheme`
Names this FMRI scheme

`version`
Scheme major version

Together these select the set of rules (as below) by which the given FMRI should be interpreted. A small number of scheme names are defined to cope with identifying resources within differing subsystems, and each scheme is versioned starting at version number 0.

When using the informal string representation of an FMRI, the scheme is always written but the scheme version in use is usually not included. The version in use is understood to be the latest for which the given FMRI string matches the FMRI-to-string conversion rules.

FMRI Authority

A further required, although implicitly present in some cases, FMRI member is `authority`:

```
Member Name   Data Type  Stability
--------------------------------------
[authority]   nvlist     Committed, (Varies)
```

Every FMRI includes authority information, either explicitly with the `authority` `nvlist` if present in the FMRI or implicitly that of the local fault management domain if not present. Authority information has two flavors:

- High-level authority information, such as to identify the platform and hostname within which a particular resource that is the subject of an FMRI is located
- Leaf-level authority information, providing identity information for the specific resource that is the subject of the FMRI

For example, an FMRI identifying a disk can have high-level authority information detailing the platform (host name, product name, chassis serial) in which the disk resides, and leaf-level authority information detailing the individual disk part number, serial, and so forth.

It is an unfortunate fact that some FMRI schemes have evolved to include leaf-level authority information outside of the `authority` `nvlist` as standalone FMRI members, and written as part of the path component in string form.

In their string representation, both high-level and leaf-level, FMRI authority information is prefaced with a colon (:), for example:

```
hc://:chassis-serial=12345/<string>
```

“:chassis-serial=12345” is a string represented authority member.

There are two versions of the high level authority (version 0 and version 1). Following are the descriptions for each.

The leaf-level authority members are described in the scheme specific sections below.

Version 0

```
Member Name   Data Type   Stability
--------------------------------------------------
[server-id]   string      Committed, Not an Interface
[chassis-id]  string      Committed, Not an Interface
```

```
[product-sn]  string      Committed, Not an Interface
[product-id]  string      Committed, Not an Interface
[domain-id]   string      Committed, Not an Interface
[host-id]     string      Committed, Not an Interface
```

`[server-id]`
The hostname (`uname -n`) string for the entity on which the fault manager is running. Not invariant.

`[chassis-id]`
This is a serial number identifying the chassis within which the resource resides. Some large systems can include multiple distinct chassis components.

`[product-sn]`
In some cases chassis-id alone does not fully identify the system, for example, for service entitlement keyed by a product serial number and where the product can comprise multiple chassis elements. Invariant.

`[product-id]`
The product name string, such as Sun-Fire-X4600. Invariant.

`[domain-id]`
Identifies a particular hardware domain in the case of hardware that supports this feature. Not invariant.

`[host-id]`
The `hostid(1)` string in Oracle Solaris); not generally used. Not necessarily invariant.

Version 1

```
Member Name     Data Type    Stability
-------------------------------------------------------
[system-mfg]      string      Committed, Not an Interface
[system-name]     string      Committed, Not an Interface
[system-part]     string      Committed, Not an Interface
[system-serial]   string      Committed, Not an Interface
[sys-comp-mfg]    string      Committed, Not an Interface
[sys-comp-name]   string      Committed, Not an Interface
[sys-comp-part]   string      Committed, Not an Interface
[sys-comp-serial] string      Committed, Not an Interface
[chassis-mfg]     string      Committed, Not an Interface
[chassis-name]    string      Committed, Not an Interface
[chassis-part]    string      Committed, Not an Interface
[chassis-serial]  string      Committed, Not an Interface
[chassis-alias]   string      Committed, Not an Interface
[server-name]     string      Committed, Not an Interface
[domain-name]     string      Committed, Not an Interface
[host-id]         string      Committed, Not an Interface
```

`[system-mfg]`
The manufacturer of the top-level product. In a complex product the system-* members contain the service entitlement data. In a simple standalone rack system the system-*, sys-comp-* and chassis-* data is duplicated. Invariant.

`[system-name]`
The model name of the top-level product. Invariant.

`[system-part]`
The part number of the top-level product. Invariant.

`[system-serial]`
The serial number of the top-level product. Invariant.

`[sys-comp-mfg]`
The manufacturer of the component system-level product. Invariant.

`[sys-comp-name]`
The model name of the component system-level product. Invariant.

`[sys-comp-part]`
The part number of the component system-level product. Invariant.

`[sys-comp-serial]`
The serial number of the component system-level product. Invariant.

`[chassis-mfg]`
The manufacturer of the chassis within which the resource resides. Some large systems can include multiple distinct chassis components. Invariant.

`[chassis-name]`
The model name of the chassis. Invariant.

`[chassis-part]`
The part number of the chassis. Invariant.

`[chassis-serial]`
The serial number of the chassis. Invariant.

In some cases a single set of *-name/*-part/*-serial alone does not fully identify the system, for example, for service entitlement keyed by the product serial number and where the product can comprise multiple chassis elements. Invariant.

`[chassis-alias]`
This is the Chassis Receptacle Occupant alias value assigned to this chassis via ‘fmadm add-alias‘. Not invariant.

`[server-name]`
The hostname (`uname -n`) string for the entity on which the fault manager is running. Not invariant.

`[domain-name]`
Identifies a particular hardware domain in the case of hardware that supports this feature. Not invariant.

`[host-id]`
The `hostid(1)` string in Oracle Solaris); not generally used. Not necessarily invariant.

Logical vs. Universal FMRI Schemes

A `logical` FMRI scheme defines FMRIs that can only meaningfully be interpreted within the fault management domain (typically an Oracle Solaris instance) in which they were generated. Identical FMRIs of a logical scheme that are native to distinct fault management domains do not necessarily identify the same actual resource. For example a `cpu` identified as `cpu id 1` using the `cpu` scheme must be interpreted in the Oracle Solaris instance that generated the FMRI (that is, its native context), and many other Oracle Solaris instances also have a logical processor id number `1` but all such FMRIs do not identify the same actual processor. The native fault management domain is identified by its high-level authority information which is typically not captured in an FMRI instance but instead understood to be that of the local Oracle Solaris instance.

A `universal` FMRI scheme identifies resources in a universally unique manner, and two identical FMRIs in a universal scheme identify the same actual resource wherever they are interpreted. Such schemes are used when ambiguity must be avoided, such as in identifying hardware components that are faulted.

FMRI schemes do not include a member indicating whether the scheme is logical or universal.

FMRI Schemes and Versions

The following FMRI scheme versions are defined in the sections that follow:

```
Scheme Version(s) Universal? Description
---------------------------------------------------------------
cpu    0, 1       No         Logical processor identification
dev    0          No         Device resources
fmd    0, 1       No         Fault Management Daemon modules
hc     0, 1       Yes        Hardware Components
mem    0          No         Memory modules and cache
mod    0          No         Kernel modules
pkg    0, 1       Yes        Software packages
svc    0          No         SMF services & service instances
sw     0          No         Software objects
zfs    0          No         zfs filesystem resources
```

Of these, the `svc` and `pkg` schemes are the most-likely to be encountered in day-to-day Oracle Solaris use, while the remainder are usually hidden behind command line interaction with the fault management subsystem when problems have been diagnosed.

SCHEME cpu VERSION 0

The cpu scheme is used to identify the logical Oracle Solaris processor resource that can be affected by a problem that has been diagnosed, and which can be offlined in response to that problem diagnosis. It does not necessarily identify the physical resource involved, and should not be used to identify parts for replacement.

The cpu scheme is a logical FMRI scheme; an FMRI in the cpu scheme is meaningful only within the fault management domain identified by the authority information.

```
Member Name   Data Type    Stability
-----------------------------------
scheme        string       Committed, value "cpu"
version       uint8        Committed, value 0
cpuid         uint32       Committed
[serial]      uint64       Private
```

cpuid
The Oracle Solaris processor instance id of this logical execution unit, as listed by psradm(1M) on the Oracle Solaris instance identified by the authority information.

[serial]
The semantics of this Private FMRI member are not documented.

String form:

```
cpu:///cpuid=<cpuid>[/serial=<hex-serial>]
```

<hex-serial> is the *serial* member written in hexadecimal with no 0x prefix.

SCHEME cpu VERSION 1

Version 1 of the cpu FMRI scheme has the same use as version 0, with some private FMRI member changes and additions.

```
Member Name   Data Type   Stability
-----------------------------------------------
scheme        string      Committed, value "cpu"
version       uint8       Committed, value 1
cpuid         uint32      Committed
[serial]      string      Private
[cpumask]     uint8       Private
[cpufru]      string      Private
[cacheindex]  uint32      Private
[cacheway]    uint32      Private
[cachebit]    uint16      Private
[cachetype]   uint8       Private

cpuid

       As per cpu scheme version 0

[serial]
```

```
[cpumask]
[cpufru]
[cacheindex]
[cacheway]
[cachebit]
[cachetype]
```

The semantics of these Private FMRI members are not documented.

`String form:`

```
cpu:///cpuid=<cpuid>[/serial=<serial>]
         [/cacheindex=.../cacheway=.../
         cachebit=.../cachetype=...]
```

Members `cpumask` and `cpufru` do not appear in the string form.

SCHEME dev VERSION 0

The `dev` scheme is used to identify devices. It is not an invariant scheme. A `dev` scheme FMRI should be interpreted only in the fault management domain identified within the authority information.

```
Member Name        Data Type   Stability
-------------------------------------------------------------------
scheme             string      Committed, value "dev"
version            uint8       Committed, value 0
device-path        string      Committed
[devid]            string      Private
[target-port-l0id] string      Private
```

`device-path`
A filesystem path within the Oracle Solaris `/devices` tree that identifies a device node, such as `pci@0,0/pci108e,cb84@2,1`. Beyond this interpretation that this member identifies a `/devices` path, the internal structure of such paths, that is, the structure and node naming of the Oracle Solaris `/devices` tree) is `Not An Interface`.

`[devid]`
`[target-port-l0id]`
The semantics of these `Private` FMRI members are not documented.

`String form:`

```
dev:///[:devid=<devid>]
    [:target-port-l0id=<target-port-l0id>]
    <device-path>
```

SCHEME fmd VERSION 0

This scheme is used to identify Oracle Solaris Fault Manager (`fmd(1M)`) modules.

High-level authority:

```
Member Name  Data Type  Stability
--------------------------------
[server-id]  string     Committed, Not an Interface
[chassis-id] string     Committed, Not an Interface
[product-sn] string     Committed, Not an Interface
[product-id] string     Committed, Not an Interface
[domain-id]  string     Committed, Not an Interface
```

FMRI does not include any leaf-level authority information within the authority nvlist, instead it is featured as top-level members of the FMRI.

```
Member Name  Data Type  Stability
---------------------------
scheme       string     Committed, value "fmd"
version      uint8      Committed, value 0
[authority]  nvlist     Committed, See above
mod-name     string     Committed, Not an Interface
mod-version  string     Committed, Not an Interface
```

`mod-name`
The `fmd` module name, an opaque string.

`mod-version`
The `fmd` module version, an opaque string.

String form:
`fmd:///module/mod-name>`

SCHEME fmd VERSION 1

This scheme is used to identify Oracle Solaris Fault Manager (`fmd(1M)`) modules.

High-level authority:

```
Member Name       Data Type  Stability
-------------------------------------
[system-mfg]      string     Committed, Not an Interface
[system-name]     string     Committed, Not an Interface
[system-part]     string     Committed, Not an Interface
[system-serial]   string     Committed, Not an Interface
[sys-comp-mfg]    string     Committed, Not an Interface
[sys-comp-name]   string     Committed, Not an Interface
[sys-comp-part]   string     Committed, Not an Interface
[sys-comp-serial] string     Committed, Not an Interface
[server-name]     string     Committed, Not an Interface
[domain-name]     string     Committed, Not an Interface
[host-id]         string     Committed, Not an Interface
```

As with version 0 the FMRI does not contain leaf-level authority informaion within the authority nvlist. Instead it is featured as top-level members of the FMRI.

```
Member Name       Data Type  Stability
-----------------------------------
scheme            string     Committed, value "fmd"
version           uint8      Committed, value 1
[authority]       nvlist     Committed, See above
mod-name          string     Committed, Not an Interface
mod-version       string     Committed, Not an Interface
```

`mod-name`
The `fmd` module name, an opaque string.

`mod-version`
The `fmd` module version, an opaque string.

String form:

```
fmd:///module/<mod-name>
```

SCHEME legacy-hc VERSION 0

The hc scheme is used to identify hardware components. In most cases these FMRIs should remain as an internal implementation detail, and where a hardware component is diagnosed as faulty it is identified (for example, in `fmadm(1M)`) by its FRU label or location label.

There is only one version of the `hc` scheme in use (version 0) but we differentiate two minor versions, distinguished by the presence or absence of a member named component. An `hc` scheme FMRI that includes a component member is an instance of the obsoleted legacy `hc` scheme.

Obsolete legacy `hc` variant:

```
Member Name  Data Type  Stability
-------------------------------------------------------------------
scheme       string     Committed, value "hc"
version      uint8      Committed, value 0
component    string     Committed
```

`component`
The FRU label for the resource.

String form for legacy `hc`:

```
hc:///component=<component>
```

SCHEME hc VERSION 0

The authority `nvlist` is the high-level authority information concerned with identifying the fault management domain in which a given FMRI was generated. Thus it includes members for the hostname, product name, chassis serial number and so on. Some of these members are invariant (such as platform serial number) and serve uniquely to identify some element, while others (such as hostname) are a soft identification which could change over time (albeit infrequently).

```
Member Name   Data Type   Stability
-----------------------------------------------------
[server-id]   string      Committed, Not an Interface
[chassis-id]  string      Committed, Not an Interface
[product-sn]  string      Committed, Not an Interface
[product-id]  string      Committed, Not an Interface
[domain-id]   string      Committed, Not an Interface
[host-id]     string      Committed, Not an Interface
```

While all members are listed as optional, as many as make sense for a given FMRI should be included. If the authority `nvlist` is absent in an FMRI then the authority is understood to be the local fault management domain; if an `hc` scheme FMRI is transmitted outside of its native domain it should include explicit high-level authority information.

The `Not an Interface` stability for the data semantics of each means that the value can be treated as an opaque string only and any internal structure to the string is not an interface. For example a serial number can be matched using string compare, but the length of the serial number string and the meaning of individual characters therein is not an interface.

String form:

These authority members, those that are present in a given FMRI, can be written in string form as an unordered series of colon-separated *name*=*value* pairs. We'll define this as `authstr`, to be used in the specification of the string form for individual FMRI schemes below which import the default authority information.

```
authstr = server-id=<server-id>
                [:chassis-id=<chassis-id>]
                [:product-sn=<product-sn>]
                [:product-id=<product-id>]
                [:domain-id=<domain-id>]
                [:host-id=<host-id>]
                [:devid=<devid>]
```

There is no leaf-level authority/identity information included in the authority `nvlist` - instead such information (serial and part numbers) feature as top-level members of the FMRI.

```
Member Name    Data Type    Stability
-----------------------------------------------------
scheme         string       Committed, value "hc"
version        uint8        Committed, value 0
[authority]    nvlist       Committed, See above
[serial]       string       Committed, Not-an-Interface
[devid]        string       Private
[part]         string       Committed, Not-an-Interface
[revision]     string       Committed, Not-an-Interface
[hc-root]      string       Private
hc-list-sz     uint32       Private
hc-list        nvlist array Private
```

```
[facility]     nvlist       Private
[hc-specific]  nvlist       Private
```

`[serial]`
The serial number string of the hardware component identified in this FMRI. This is an opaque string whose internal structure is not an interface.

`[devid]`
The semantics of these Private FMRI members are not documented.

`[part]`
The part number string of the hardware component identified in this FMRI. This is an opaque string whose internal structure is not an interface.

`[revision]`
The revision of the part number, as a string. Again this is an opaque string.

`[hc-root]`
The semantics of this Private FMRI member are not documented.

`hc-list-sz`
The semantics of this Private FMRI member are not documented.

`hc-list`
The semantics of this Private FMRI member are not documented, however we enumerate the structure of the content (subject to incompatible change). Each member of the (ordered) `nvlist` array has the following structure:

```
Member Name    Data Type   Stability
-----------------------------------
hc-name        string      Private
hc-id          string      Private
```

`[facility]`
The semantics of this Private FMRI member are not documented, however we enumerate the structure of the content (subject to incompatible change). The facility `nvlist` has members as follows:

```
Member Name    Data Type   Stability
-----------------------------------
facility-type  string      Private
facility-name  string      Private
```

`[hc-specific]`
The semantics of this Private FMRI member are not documented.

`String form:`

```
authstr is defined above
hcauthstr = [:serial=<serial>]
                [:devid=<devid>]
                [:part=<part>]
```

```
                [:revision=<revision>]

hc-path =<hc-list[0].hc-name>=<hc-list[0].hc-id>
               /<hc-list[1].hc-name>=<hc-list[1].hc-id>
               /<hc-list[2].hc-name>=<hc-list[2].hc-id>
               ...
hcspecstr = private string formed from hc-specific nvlist,
               of the form "/name=value"

facstr = <facility.facility-type>=<facility.facility-name>

hc://[authstr][hcauthstr]/[hc-root/]<hc-path>
                                    [<hcspecstr>]
                                    [?<facstr>]
```

The high-level authority information from the authority nvlist appears adjacent to the leaf-level authority information of serial and part number, all colon separated. The components of this authority string can appear in any order, with high-level and leaf authority elements interleaved. Parsing such a string is clumsy and relies on knowing in advance which components belong to which class of authority information.

SCHEME hc VERSION 1

Version 1 of the hc FMRI scheme has the same use as version 0, with some private FMRI member changes and additions.

```
Member Name     Data Type    Stability
-------------------------------------------------------
[chassis-mfg]     string      Committed, Not an Interface
[chassis-name]    string      Committed, Not an Interface
[chassis-part]    string      Committed, Not an Interface
[chassis-serial]  string      Committed, Not an Interface
[chassis-alias]   string      Committed, Not an Interface
```

String form:

The same semantics apply to the hc FMRI scheme version 1 string form as version 0.

```
authstr = [:chassis-mfg=<chassis-mfg>]
                [:chassis-name=<chassis-name>]
                [:chassis-part=<chassis-part>]
                [:chassis-serial=<chassis-serial>]
                [:chassis-alias=<chassis-alias>]
```

As with hc scheme version 0 there is no leaf-level authority/identity information included in the authority nvlist with version 1 - instead such information (for example, serial/revision/part numbers) feature as top-level members of the FMRI.

```
Member Name    Data Type    Stability
-------------------------------------------------------
scheme         string       Committed, value "hc"
```

```
version         uint8        Committed, value 1
[authority]     nvlist       Committed, See above
[fru-mfg]       string       Committed, Not-an-Interface
[fru-serial]    string       Committed, Not-an-Interface
[fru-part]      string       Committed, Not-an-Interface
[fru-revision] string        Committed, Not-an-Interface
[devid]         string       Private
[hc-root]       string       Private
hc-list-sz      uint32       Private
hc-list         nvlist array Private
[facility]      nvlist       Private
[hc-specific]   nvlist       Private
```

`[fru-mfg]`
The manufacturer string of the hardware component identified in this FMRI. This is an opaque string whose internal structure is not an interface.

`[fru-serial]`
The serial number string of the hardware component identified in this FMRI. This is an opaque string whose internal structure is not an interface.

`[fru-part]`
The part number string of the hardware component identified in this FMRI.This is an opaque string whose internal structure is not an interface.

`[fru-revision]`
The revision of the part number, as a string. Again this is an opaque string.

`[devid]`
The semantics of these Private FMRI members are not documented.

`[hc-root]`
The semantics of this Private FMRI member are not documented.

`hc-list-sz`
The semantics of this Private FMRI member are not documented.

`hc-list`
The semantics of this Private FMRI member are not documented, however we enumerate the structure of the content (subject to incompatible change). Each member of the (ordered) nvlist array has the following structure:

```
Member Name    Data Type   Stability
-----------------------------------
hc-name        string      Private
hc-id          string      Private
```

`[facility]`
The semantics of this Private FMRI member are not documented, however we enumerate the structure of the content (subject to incompatible change). The facility nvlist has members as follows:

```
Member Name    Data Type   Stability
-----------------------------------
facility-type  string      Private
facility-name  string      Private
```

[hc-specific]
The semantics of this Private FMRI member are not documented.

String form:

```
authstr is defined above
hcauthstr = [:fru-mfg=<fru-mfg>]
                [:fru-serial=<fru-serial>]
                [:devid=<devid>]
                [:fru-part=<fru-part>]
                [:fru-revision=<fru-revision>]

hc-path =<hc-list[0].hc-name>=<hc-list[0].hc-id>
              /<hc-list[1].hc-name>=<hc-list[1].hc-id>
              /<hc-list[2].hc-name>=<hc-list[2].hc-id>
              ...
hcspecstr = private string formed from hc-specific nvlist,
              of the form "/name=value"

facstr = <facility.facility-type>=<facility.facility-name>

hc://[authstr][hcauthstr]/[hc-root/]<hc-path>
                                    [<hcspecstr>]
                                    [?<facstr>]
```

The high-level authority information from the authority nvlist appears adjacent to the leaf-level authority information of serial and part number, all colon separated. The components of this authority string can appear in any order, with high-level and leaf authority elements interleaved. Parsing such a string is clumsy and relies on knowing in advance which components belong to which class of authority information.

SCHEME mem VERSION 0

This scheme is used to identify memory modules and memory pages, although more recently these have been identified in the hc scheme. The mem scheme has also been used to identify cpu cache resources.

```
Member Name   Data Type    Stability
-----------------------------------------------
scheme        string       Committed, value "mem"
version       uint8        Committed, value 0
unum          string       Committed, Private
[serial]      string array Private
[physaddr]    uint64       Private
[offset]      uint64       Private
```

`unum`
This string identifies the memory module, generally as labelled on the platform silkscreens or service label. The presence of this member of type string is a Committed interface, but the internal structure of the `unum` string is a `Private` interface. Therefore, you can use the opaque `unum` string to identify a memory module, but the interpretation of the components of a `unum` string is platform dependent and `Private`.

A `unum` can identify a group (or bank) of memory modules. These should instead have been identified by multiple FMRIs, one for each memory module, but this is an historical anomaly from early implementations.

`[serial]`
A string array of the serial number(s) of the memory module(s) named by the `unum` member. The internal structure of a serial number string is not an interface so you can use only the opaque string value in matching serial number and not apply any further interpretation.

`[physaddr]`
`[offset]`
The semantics of these Private FMRI members are not documented.

`String form:`

```
mem:///unum=<unum>
        [/offset=<hex-offset> | /physaddr=<hex-physaddr>]
```

SCHEME mod VERSION 0

This scheme is used to identify Oracle Solaris kernel modules.

Member Name	Data Type	Stability
scheme	string	Committed, value "fmd"
version	uint8	Committed, value 0
mod-name	string	Committed
[mod-desc]	string	Committed
[mod-pkg]	fmri	(fmri)

`mod-name`
The module name, as in per `modinfo(1M)`.

`[mod-desc]`
The module description, again as per `modinfo(1M)`.

`[mod-pkg]`
A `pkg` scheme FMRI of the package which delivers this module.

`String form:`

```
mod:///mod-name=<mod-name>
```

SCHEME pkg VERSION 0

This version of the pkg scheme corresponds to legacy SVR4 packaging as used with pkgadd(1M), pkginfo(1), pkgrm(1M), and related utilities. This scheme is universal if package name and package version conventions are adhered to.

```
Member Name   Data Type   Stability
-----------------------------------------------------------------
scheme        string      Committed, value "pkg"
version       int8        Committed, value 0
pkg-name      string      Committed
pkg-basedir   string      Committed
pkg-inst      string      Committed
pkg-version   string      Committed
```

pkg-name
The package name.

pkg-basedir
BASEDIR as per pkginfo(1)

pkg-inst
PKGINST as per pkginfo(1)

pkg-version
VERSION as per pkginfo(1)

String form:

```
pkg:///<pkg-name>/:version=<pkg-version>
```

SCHEME pkg VERSION 1

This version of the pkg scheme corresponds to the pkg(5) Image Packaging System. Such FMRIs are used in string form in the pkg(1) command line and its output, and the command line permits abbreviations of the FMRI string.

FMRIs in the pkg scheme version 1 are universal: the same FMRI interpreted in two distinct contexts (such as in distinct Oracle Solaris instances) identify the same actual package (or copies thereof).

The pkg(5) command line uses the string form of version 1 pkg FMRIs, and permits abbreviations of such strings.

```
Member Name   Data Type    Stability
-------------------------------------------------
scheme        string       Committed, value "pkg"
version       uint8        Committed, value 1
[authority]   nvlist       Committed
pkg-name      string       Committed
[pkg-version] nvlist       Committed
```

[authority]
The authority nvlist specifies the package publisher, per pkg publisher output:

```
Member Name    Data Type    Stability
-----------------------------------
publisher      string       Committed
```

`publisher`

The publisher of this package, such as Oracle Solaris.

`pkg-name`

The name (stem) of the package

`[pkg-version]`

```
Member Name    Data Type    Stability
---------------------------------------
release        string       Committed
[built-on]     string       Committed
[branch]       string       Committed
[timestamp]    string       Committed
```

`release`

The primary version of the component, written as a dot sequence string (an arbitrary-length, dot-separated series of non-zero-leading decimal digits).

`built-on`

The operating system version on which the component was built, again specified as a dot sequence.

`branch`

The vendor branch version, as a dot sequence.

`timestamp`

An ISO 8601 date string: *yyyymmdd*`T`*hhmmss*`Z` in which the `T` and `Z` are literals.

`String form:`

```
pkgauthstr = <authority.publisher>

pkg://[pkgauthstr]/<pkg-name>
    [@<release>[,<built-on>][-<branch>][:<timestamp>]
```

SCHEME svc VERSION 0

The `svc` scheme describes SMF (see `smf(5)` services and service instances. In SMF command lines such as `svcadm(1M)`, `svccfg(1M)`, `svcprop(1)`, `svcs(1)` and their output such FMRI are always written in their string form, and the SMF command lines permit abbreviations.

The `svc` scheme FMRI is a logical scheme, and, as such an FMRI should only be interpreted in the fault management domain (Oracle Solaris instance) in which it was generated.

```
Member Name    Data Type   Stability
-------------------------------------------------
scheme         string      Committed, value "svc"
```

```
version         uint8       Committed, value 0
svc-name        string      Committed
[svc-instance] string       Committed
[contract-id]   string      Committed
[svc-scope]     string      Committed
```

```
pkg://[pkgauthstr]/<pkg-name>
```

svc-name
The service name, such as network/smtp. See smf(5).

[svc-instance]
Used when identifying an instance of a service. Often default, but can also be things like the sendmail instance of service network/smtp.

[contract-id]
This identifies the numeric (although written as a decimal string) contract id (see smf(5) and contract(4) used to manage a particular running instance of a service.

[svc-scope]
As per smf(5), all service and instance objects are contained in a scope that represents a collection of configuration information. The configuration of the local Oracle Solaris instance is called the localhost scope, and is the only currently supported scope.

String form:

```
svc://[<svc-scope>]/<svc-name>
        [:<svc-instance>[@<contract-id>]]
```

SMF subsystem commands such as svcadm(1M) permit (or even require) abbreviations of the string form above. The indication of svc scheme at the start of the string is considered optional and implied by the SMF command line utility. When the svc scheme identifier is used, the authority component is also abbreviated to svc:/<svc-name> instead of the more-formal svc:///<svc-name>. For example, in SMF we write

```
svc:/network/smtp:sendmail
```

instead of the more-formal

```
svc:///network/smtp:sendmail
```

or even the more formal

```
svc://localhost/network/smtp:sendmail
```

Furthermore, SMF permits further abbreviation if it identifies a unique service or instance. For example the instance name sendmail is unique and one can use svcs sendmail, svcadm restart sendmail, and so forth.

Such abbreviations are a convention of the SMF subsystem and not part of the formal FMRI definition. When a svc scheme FMRI is received in nvlist form (for example, in a libfmevent(3LIB) subscription it appears in the more formal form when rendered as a

string.

SCHEME sw VERSION 0

This FMRI scheme is used to identify software resources such as executables, library objects, and core files. This is a logical FMRI scheme.

```
Member Name   Data Type   Stability
-----------------------------------------------
scheme        string      Committed, value "sw"
version       uint8       Committed, value 0
object        nvlist      See belowPrivate
[site]        nvlist      See belowPrivate
[context]     nvlist      See belowPrivate
```

The `object`, `site`, and `context` members are all `Private` and so subject to incompatible change, but their content is documented below for informational purposes.

`object`
This required member identifies the software object. At this time only objects resident within the filesystem are catered for.

```
Member     Type        Description
-------------------------------------------------------------
path       string      Filesystem path to object
[root]     string      If present, real path to chroot root
                       directory
[pkg]      fmri        "pkg" scheme fmri of package that
                       delivers path
```

`site`
This optional member identifies a site within the object.

```
Member   Type    Description
-------------------------------------------------------------
[token]  string  Vendor and subsystem unique publisher
                 token id
[module] string  Source module information
[file]   string  Source filename (translation unit)
[func]   string  Source function
[line]   int64   Source file line number
```

`context`
This optional member communicates runtime information.

```
Member             Type         Description
-----------------------------------------------------
[origin]           string       "userland" or "kernel"
[execname]         string       Executable name
[pid]              uint64       Process id
[thread-id]        uint64       Thread id
```

```
[os-instance-uuid] string       Solaris instance UUID
[zone]             string       Zone name, if not the global zone
[ctid]             uint64       Contract id
[stack]            string array Symbolic stack trace
```

String form:

```
sw://[<authority>]/
       [:root=<object.root]
       :path=<object.path>
       [#<fragment-identifier>]
```

Slash characters / in the root and object path are not escaped.

<fragment-identifier> is one of:

```
:token=<site.token>
```

or

```
:file=<site.file>[:func=<site.func>][:line=<site.line>]
```

SCHEME zfs VERSION 0

```
Member Name   Data Type    Stability
------------------------------------------------------------------
scheme        string       Committed, value "zfs"
version       uint8        Committed, value 0
[pool-name]   string       Committed
pool          uint64       Committed
[vdev]        uint64       Private
[vdev-name]   string       Private
```

[pool-name]
The pool name, as per zpool list -o name

pool
The pool GUID as per zpool list -o guid

[vdev]
The semantics of this Private FMRI member are not documented

[vdev-name]
The semantics of this Private FMRI member are not documented.

String form:

```
hex-vdev = 'vdev' in hexadecimal with no '0x' prefix.
hex-pool = 'pool' in hexadecimal with no '0x' prefix.
zfs://[pool_name=<pool-name>/]pool=<hex-pool>
zfs://[pool_name=<zfs://[pool_name=<hex-pool>
[/vdev=<hex-vdev>][:vdev_name=<vdev-name>]
```

The <vdev-name> is written without escaping any / characters.

関連項目

hostid(1), fmadm(1M), fmd(1M), fmdump(1M), pkgadd(1M), pkginfo(1), pkgrm(1M), psradm(1M), svcadm(1M), svccfg(1M), svcprop(1), svcs(1), libfmevent(3LIB), libnvpair(3LIB), contract(4), attributes(5), smf(5)

pkg(5)

RFC 2396

名前　fnmatch – ファイル名のパターンマッチング

機能説明　次に説明するパターンマッチング表記法を使用して、シェル内で一致する文字列のパターンを指定します。従来のパターンマッチング表記法も正規表現に対応していますが、若干異なっています。このため、このパターンマッチング表記法の規則については、regex(5) のマニュアルページで記述されている正規表現表記法に基づいて説明します。

単一の文字を照合するパターン　次の単一の文字を照合するパターンは、単一の文字 (通常の文字、特殊パターン文字、パターンブラケット表現) を照合します。パターンブラケット表現は、単一の照合要素もマッチングします。

通常の文字とは、その文字自体を照合するパターンです。NUL、引用符を必要とする特殊なシェル文字、および次の 3 つの特殊パターン文字を除いて、サポートされる文字セットに任意の文字を指定できます。マッチングは文字のグラフィック表現ではなく、文字のエンコーディングに使用されるビットパターンに基づいて行なわれます。任意の文字 (通常、シェル特殊、特殊パターン) に引用符を付けると、そのパターンはその文字自体を照合します。シェル特殊文字には、常に引用符が必要です。

引用符なしで、ブラケット表現以外の次の 3 つの文字には、パターン指定において特別な意味があります:

?　疑問符は任意の文字を照合するパターンです。

*　後述の「複数の文字を照合するパターン」で説明するように、アスタリスクは複数の文字を照合するパターンです。

[　開始括弧から、パターンブラケット表現が導入されます。

regex(5) のマニュアルページに記述された基本正規表現におけるブラケット表現についての説明は、正規表現表記法では一致しないリストの役割において感嘆符 (!) がサーカムフレックス文字 (^) に置換される点を除いて、パターンブラケット表現にも当てはまります。引用符なしのサーカムフレックス文字でブラケット表現が始まる場合は、不特定の結果が発生します。

ブラケット表現でサーカムフレックスを制限すると、感嘆符に加えてサーカムフレックスを否定文字として使用するパターンマッチングをサポートする実装が可能になります。移植性のあるアプリケーションでどちらかの文字をマッチングするには、[\^!] などを使用する必要があります。

シェルの引用符が削除されていない (exec 関数のいずれかを使用して find が呼び出されるときの find –name 一次子への引数や、fnmatch(3C) 関数への *pattern* 引数など) パターンマッチングが使用されている場合は、特殊文字の前にバックスラッシュ文字を付けてエスケープすれば、特殊な意味を取り除くことができます。このようなエスケープするバックスラッシュは破棄されます。シーケンス \\ は 1 つのリテラ

ルバックスラッシュを表します。通常の文字、特殊なシェル文字、および特殊なパターン文字に引用符を付ける要件および効果はすべて、このコンテキストのエスケープに適用されます。

パターンマッチングは 3 つの別々の状況で動作する必要があるため、ここでは引用符とエスケープの両方について説明します。

- シェルで直接呼び出す (パス名の展開や `case` 文など)。次のパターンはすべて、文字列またはファイル `abc` を照合します。

`abc`	`"abc"`	`a"b"c`	`a\bc`	`a[b]c`
`a["b"]c`	`a[\b]c`	`a["\b"]c`	`a?c`	`a*c`

  次は照合しません:

`"a?c"`	`a\*c`	`a\[b]c`

- シェルを介さずにユーティリティーや関数を呼び出す (`find(1)` および関数 `fnmatch(3C)` を参照)。
- シェルコマンド行を介して `find`、`cpio`、`tar`、`pax` などのユーティリティーを呼び出す。この場合、ユーティリティーで引数が現れる前に、シェルの引用符が削除されます。例:

  find /bin -name e\c[\h]o -print

  引用符が削除されたあとに、バックスラッシュが `find` に渡され、エスケープ文字として処理されます。どちらも通常の文字の前に付加されているため、`c` と `h` はその文字自体を意味し、`echo` は (`/bin` にある) 従来の多くのシステムで見つかります。特殊なシェル文字やパターン文字が含まれるファイル名を検索するには、次のように、引用符とエスケープ文字の両方が必要です:

  ```
  pax -r . . . "*a\ ( \?"
  ```

  この目的は、`a(?` で終わるファイル名を抽出することです。

特殊なシェル文字 (メタ文字とも呼ばれる) に引用符やエスケープ文字を付けるには、規格に準拠するアプリケーションが必要です。このような保護なしで使用すると、構文エラーが発生したり、実装の拡張機能がトリガーされたりする可能性があります。たとえば、KornShell のパターンでは、括弧に基づく一連の拡張機能がサポートされています。`ksh(1)` を参照してください。

複数の文字を照合するパターン

次の規則は、単一の文字を照合するパターンから複数の文字を照合するパターンを構成する際に使用されます。

- アスタリスク (*) は、null 文字列を含む任意の文字列を照合するパターンです。

- 単一の文字を照合するパターンの連結は、単一の文字を連結したものや各連結パターンで一致する照合要素と一致する有効なパターンです。
- 1 つ以上の単一の文字を照合するパターンと 1 つ以上のアスタリスクとの連結は、有効なパターンです。このようなパターンでは、各アスタリスクは 0 文字以上の文字列を照合し、パターンの残りの部分が文字列と最長一致する文字を照合できます。

各アスタリスクは 0 回以上の繰り返しを照合するため、パターン `a*b` と `a**b` のはたらきはまったく同じです。

たとえば、

`a[bc]` 文字列 `ab` および `ac` に一致します。

`a*d` 文字列 `ad`、`abd`、および `abcd` と一致しますが、文字列 `abc` とは一致しません。

`a*d*` 文字列 `ad`、`abcd`、`abcdef`、`aaaad`、および `adddd` と一致します。

`*a*d` 文字列 `ad`、`abcd`、`efabcd`、`aaaad`、および `adddd` と一致します。

ファイル名の展開に使用されるパターン

これまでに「複数の文字を照合するパターン」および「単一の文字を照合するパターン」で説明した規則は、パターンマッチング表記法がファイル名の展開で使用されるときに適用される次の規則によって限定されます。

1. パス名のスラッシュ文字は、パターン内で 1 つ以上のスラッシュを使用して明示的に照合する必要があり、アスタリスクや疑問符などの特殊文字またはブラケット表現で照合することはできません。パターン内のスラッシュは、ブラケット表現の前に識別されるため、ファイル名の展開に使用されるパターンブラケット表現にスラッシュを含めることはできません。たとえば、パターン `a[b/c]d` は、`abd` や `a/d` などのパス名と一致します。文字どおり `a[b/c]d` のパス名のみと一致します。
2. ファイル名がピリオド (.) から始まる場合は、パターンの最初の文字としてピリオドを使用するか、直後にスラッシュ文字を付けて、ピリオドを明示的に照合する必要があります。次のパターンでは、先頭のピリオドは照合されません:

   • アスタリスクや疑問符などの特殊文字

   • 次のような一致しないリストを含むブラケット表現:

   `[!a]`

   次のような範囲表現:

   `[%–0]`

   または、次のような文字クラス表現:

   `[[:punct:]]`

   次のように、ブラケット表現マッチングリスト内の明示的なピリオドが、

`[.abc]`

ファイル名の先頭ピリオドを照合できるかどうかは指定されません。

3. 必要に応じて、指定されたパターンは既存のファイル名およびパス名と照合されます。各コンポーネントにパターン文字が含まれている場合は、そのコンポーネントを含むディレクトリに読み取り権限が必要です。コンポーネント (最後のコンポーネントを除く) にパターン文字が含まれていない場合は、検索権限が必要です。たとえば、次のパターンがあるとします:

   `/foo/bar/x*/bam`

   ディレクトリ `/` および `foo` には検索権限、ディレクトリ `bar` には検索権限および読み取り権限、各 `x*` ディレクトリには検索権限が必要です。

   パターンが既存のファイル名またはパス名と一致すると、それらのファイル名とパス名に置き換えられ、現在のロケールで有効な照合順序に従ってソートされます。パターンに不正なブラケット表現が含まれている場合や、既存のファイル名やパス名と一致しない場合は、パターン文字列は変更されません。

関連項目 `find(1)`、`ksh(1)`、`fnmatch(3C)`、`regex(5)`

名前

formats – file format notation

機能説明

Utility descriptions use a syntax to describe the data organization within files—stdin, stdout, stderr, input files, and output files—when that organization is not otherwise obvious. The syntax is similar to that used by the `printf(3C)` function. When used for stdin or input file descriptions, this syntax describes the format that could have been used to write the text to be read, not a format that could be used by the `scanf(3C)` function to read the input file.

Format

The description of an individual record is as follows:

"<format>", [<*arg1*>, <*arg2*>, . . ., <*argn*>]

The `format` is a character string that contains three types of objects defined below:

characters	Characters that are not *escape sequences* or *conversion specifications*, as described below, are copied to the output.
escape sequences	Represent non-graphic characters.
conversion specifications	Specifies the output format of each argument. (See below.)

The following characters have the following special meaning in the format string:

„ (An empty character position.) One or more blank characters.

/\ Exactly one space character.

The notation for spaces allows some flexibility for application output. Note that an empty character position in `format` represents one or more blank characters on the output (not *white space*, which can include newline characters). Therefore, another utility that reads that output as its input must be prepared to parse the data using `scanf(3C)`, `awk(1)`, and so forth. The character is used when exactly one space character is output.

Escape Sequences

The following table lists escape sequences and associated actions on display devices capable of the action.

Sequence	Character	Terminal Action
\\	backslash	None.
\a	alert	Attempts to alert the user through audible or visible notification.
\b	backspace	Moves the printing position to one column before the current position, unless the current position is the start of a line.
\f	form-feed	Moves the printing position to the initial printing position of the next logical page.

Sequence	Character	Terminal Action
\n	newline	Moves the printing position to the start of the next line.
\r	carriage-return	Moves the printing position to the start of the current line.
\t	tab	Moves the printing position to the next tab position on the current line. If there are no more tab positions left on the line, the behavior is undefined.
\v	vertical-tab	Moves the printing position to the start of the next vertical tab position. If there are no more vertical tab positions left on the page, the behavior is undefined.

Conversion Specifications

Each conversion specification is introduced by the percent-sign character (%). After the character %, the following appear in sequence:

flags — Zero or more *flags*, in any order, that modify the meaning of the conversion specification.

field width — An optional string of decimal digits to specify a minimum *field width*. For an output field, if the converted value has fewer bytes than the field width, it is padded on the left (or right, if the left-adjustment flag (–), described below, has been given to the field width).

precision — Gives the minimum number of digits to appear for the d, o, i, u, x or X conversions (the field is padded with leading zeros), the number of digits to appear after the radix character for the e and f conversions, the maximum number of significant digits for the g conversion; or the maximum number of bytes to be written from a string in s conversion. The precision takes the form of a period (.) followed by a decimal digit string; a null digit string is treated as zero.

conversion characters — A conversion character (see below) that indicates the type of conversion to be applied.

flags

The *flags* and their meanings are:

– The result of the conversion is left-justified within the field.

+ The result of a signed conversion always begins with a sign (+ or –).

<*space*> If the first character of a signed conversion is not a sign, a space character is prefixed to the result. This means that if the space character and + flags both appear, the space character flag is ignored.

The value is to be converted to an alternative form. For c, d, i, u, and s conversions, the behaviour is undefined. For o conversion, it increases the

precision to force the first digit of the result to be a zero. For x or X conversion, a non-zero result has 0x or 0X prefixed to it, respectively. For e, E, f, g, and G conversions, the result always contains a radix character, even if no digits follow the radix character. For g and G conversions, trailing zeros are not removed from the result as they usually are.

0 For d, i, o, u, x, X, e, E, f, g, and G conversions, leading zeros (following any indication of sign or base) are used to pad to the field width; no space padding is performed. If the 0 and – flags both appear, the 0 flag is ignored. For d, i, o, u, x and X conversions, if a precision is specified, the 0 flag is ignored. For other conversions, the behaviour is undefined.

Conversion Characters

Each conversion character results in fetching zero or more arguments. The results are undefined if there are insufficient arguments for the format. If the format is exhausted while arguments remain, the excess arguments are ignored.

The *conversion characters* and their meanings are:

d,i,o,u,x,X The integer argument is written as signed decimal (d or i), unsigned octal (o), unsigned decimal (u), or unsigned hexadecimal notation (x and X). The d and i specifiers convert to signed decimal in the style `[−]`*dddd*. The x conversion uses the numbers and letters 0123456789abcdef and the X conversion uses the numbers and letters 0123456789ABCDEF. The *precision* component of the argument specifies the minimum number of digits to appear. If the value being converted can be represented in fewer digits than the specified minimum, it is expanded with leading zeros. The default precision is 1. The result of converting a zero value with a precision of 0 is no characters. If both the field width and precision are omitted, the implementation may precede, follow or precede and follow numeric arguments of types d, i and u with blank characters; arguments of type o (octal) may be preceded with leading zeros.

The treatment of integers and spaces is different from the `printf(3C)` function in that they can be surrounded with blank characters. This was done so that, given a format such as:

`"%d\n",`<*foo*>

the implementation could use a `printf()` call such as:

`printf("%6d\n",` *foo*`);`

and still conform. This notation is thus somewhat like `scanf()` in addition to `printf( ).`

f The floating point number argument is written in decimal notation in the style `[−]`*ddd.ddd*, where the number of digits after the radix character (shown here as a decimal point) is equal to the *precision* specification. The `LC_NUMERIC` locale category determines the radix character to use in this format. If the

precision is omitted from the argument, six digits are written after the radix character; if the *precision* is explicitly 0, no radix character appears.

e,E — The floating point number argument is written in the style `[−]`*d*.*ddd*e±`dd` (the symbol ± indicates either a plus or minus sign), where there is one digit before the radix character (shown here as a decimal point) and the number of digits after it is equal to the precision. The `LC_NUMERIC` locale category determines the radix character to use in this format. When the precision is missing, six digits are written after the radix character; if the precision is 0, no radix character appears. The E conversion character produces a number with E instead of e introducing the exponent. The exponent always contains at least two digits. However, if the value to be written requires an exponent greater than two digits, additional exponent digits are written as necessary.

g,G — The floating point number argument is written in style f or e (or in style E in the case of a G conversion character), with the precision specifying the number of significant digits. The style used depends on the value converted: style g is used only if the exponent resulting from the conversion is less than −4 or greater than or equal to the precision. Trailing zeros are removed from the result. A radix character appears only if it is followed by a digit.

c — The integer argument is converted to an `unsigned char` and the resulting byte is written.

s — The argument is taken to be a string and bytes from the string are written until the end of the string or the number of bytes indicated by the *precision* specification of the argument is reached. If the precision is omitted from the argument, it is taken to be infinite, so all bytes up to the end of the string are written.

% — Write a % character; no argument is converted.

In no case does a non-existent or insufficient *field width* cause truncation of a field; if the result of a conversion is wider than the field width, the field is simply expanded to contain the conversion result. The term *field width* should not be confused with the term *precision* used in the description of %s.

One difference from the C function `printf()` is that the l and h conversion characters are not used. There is no differentiation between decimal values for type `int`, type `long`, or type `short`. The specifications %d or %i should be interpreted as an arbitrary length sequence of digits. Also, no distinction is made between single precision and double precision numbers (`float` or `double` in C). These are simply referred to as floating point numbers.

Many of the output descriptions use the term `line`, such as:

"`%s`", <*input line*>

Since the definition of `line` includes the trailing newline character already, there is no need to include a `\n` in the format; a double newline character would otherwise result.

使用例

例 1 To represent the output of a program that prints a date and time in the form Sunday, July 3, 10:02, where <*weekday*> and <*month*> are strings:

```
"%s,/\%s/\%d,/\%d:%.2d\n",<weekday>,<month>,<day>,<hour>,<min>
```

例 2 To show pi written to 5 decimal places:

```
"pi/\=/\%.5f\n",<value of pi>
```

例 3 To show an input file format consisting of five colon-separated fields:

```
"%s:%s:%s:%s:%s\n",<arg1>,<arg2>,<arg3>,<arg4>,<arg5>
```

関連項目

awk(1), printf(1), printf(3C), scanf(3C)

名前

fsattr – extended file attributes

機能説明

Attributes are logically supported as files within the file system. The file system is therefore augmented with an orthogonal name space of file attributes. Any file (including attribute files) can have an arbitrarily deep attribute tree associated with it. Attribute values are accessed by file descriptors obtained through a special attribute interface. This logical view of "attributes as files" allows the leveraging of existing file system interface functionality to support the construction, deletion, and manipulation of attributes.

The special files "." and ". ." retain their accustomed semantics within the attribute hierarchy. The "." attribute file refers to the current directory and the ". ." attribute file refers to the parent directory. The unnamed directory at the head of each attribute tree is considered the "child" of the file it is associated with and the ". ." file refers to the associated file. For any non-directory file with attributes, the ". ." entry in the unnamed directory refers to a file that is not a directory.

Conceptually, the attribute model is fully general. Extended attributes can be any type of file (doors, links, directories, and so forth) and can even have their own attributes (fully recursive). As a result, the attributes associated with a file could be an arbitrarily deep directory hierarchy where each attribute could have an equally complex attribute tree associated with it. Not all implementations are able to, or want to, support the full model. Implementation are therefore permitted to reject operations that are not supported. For example, the implementation for the UFS file system allows only regular files as attributes (for example, no sub-directories) and rejects attempts to place attributes on attributes.

The following list details the operations that are rejected in the current implementation:

`link` Any attempt to create links between attribute and non-attribute space is rejected to prevent security-related or otherwise sensitive attributes from being exposed, and therefore manipulable, as regular files.

`rename` Any attempt to rename between attribute and non-attribute space is rejected to prevent an already linked file from being renamed and thereby circumventing the `link` restriction above.

`mkdir`
symlink
mknod Any attempt to create a "non-regular" file in attribute space is rejected to reduce the functionality, and therefore exposure and risk, of the initial implementation.

The entire available name space has been allocated to "general use" to bring the implementation in line with the NFSv4 draft standard [NFSv4]. That standard defines "named attributes" (equivalent to Solaris Extended Attributes) with no naming restrictions. All Sun applications making use of opaque extended attributes will use the prefix "SUNW".

Shell-level API

The command interface for extended attributes is the set of applications provided by Solaris for the manipulation of attributes from the command line. This interface consists of a set of existing utilities that have been extended to be "attribute-aware", plus the `runat` utility designed to "expose" the extended attribute space so that extended attributes can be manipulated as regular files.

The `-@` option enable utilities to manipulate extended attributes. As a rule, this option enables the utility to enter into attribute space when the utility is performing a recursive traversal of file system space. This is a fully recursive concept. If the underlying file system supports recursive attributes and directory structures, the `-@` option opens these spaces to the file tree-walking algorithms.

The following utilities accommodate extended attributes (see the individual manual pages for details):

`cp` By default, `cp` ignores attributes and copies only file data. This is intended to maintain the semantics implied by `cp` currently, where attributes (such as owner and mode) are not copied unless the `-p` option is specified. With the `-@` (or `-p`) option, `cp` attempts to copy all attributes along with the file data.

`cpio` The `-@` option informs `cpio` to archive attributes, but by default `cpio` ignores extended attributes. See `Extended Archive Formats` below for a description of the new archive records.

`du` File sizes computed include the space allocated for any extended attributes present.

`find` By default, `find` ignores attributes. The `-xattr` expression provides support for searches involving attribute space. It returns true if extended attributes are present on the current file.

`fsck` The `fsck` utility manages extended attribute data on the disk. A file system with extended attributes can be mounted on versions of Solaris that are not attribute-aware (versions prior to Solaris 9), but the attributes will not be accessible and `fsck` will strip them from the files and place them in `lost+found`. Once the attributes have been stripped the file system is completely stable on Solaris versions that are not attribute-aware, but would now be considered corrupted on attribute-aware versions of Solaris. The attribute-aware `fsck` utility should be run to stabilize the file system before using it in an attribute-aware environment.

`fsdb` This `fsdb` utility is able to find the inode for the "hidden" extended attribute directory.

`ls` The `ls -@` command displays an "@" following the mode information when extended attributes are present. More precisely, the output line for a given file contains an "@" character following the mode characters if the `pathconf(2)` variable `XATTR_EXISTS` is set to true. See the `pathconf()` section below. The `-@` option uses the same general output format as the `-l` option.

`mv` When a file is moved, all attributes are carried along with the file rename. When a file is moved across a file system boundary, the copy command invoked is similar to the `cp -p` variant described above and extended attributes are "moved". If the extended file attributes cannot be replicated, the move operation fails and the source file is not removed.

`pax` The `-@` option informs `pax` to archive attributes, but by default `pax` ignores extended attributes. The `pax(1)` utility is a generic replacement for both `tar(1)` and `cpio(1)` and is able to produce either output format in its archive. See `Extended Archive Formats` below for a description of the new archive records.

`tar` In the default case, `tar` does not attempt to place attributes in the archive. If the `-@` option is specified, however, `tar` traverses into the attribute space of all files being placed in the archive and attempts to add the attributes to the archive. A new record type has been introduced for extended attribute entries in `tar` archive files (the same is true for `pax` and `cpio` archives) similar to the way ACLs records were defined. See `Extended Archive Formats` below for a description of the new archive records.

There is a class of utilities (`chmod`, `chown`, `chgrp`) that one might expect to be modified in a manner similar to those listed above. For example, one might expect that performing `chmod` on a file would not only affect the file itself but would also affect at least the extended attribute directory if not any existing extended attribute files. This is not the case. The model chosen for extended attributes implies that the attribute directory and the attributes themselves are all file objects in their own right, and can therefore have independent file status attributes associated with them (a given implementation cannot support this, for example, for intrinsic attributes). The relationship is left undefined and a fine-grained control mechanism (`runat(1)`) is provided to allow manipulation of extended attribute status attributes as necessary.

The `runat` utility has the following syntax:

`runat` *filename* [*command*]

The `runat` utility executes the supplied command in the context of the "attribute space" associated with the indicated file. If no command argument is supplied, a shell is invoked. See `runat(1)` for details.

Application-level API

The primary interface required to access extended attributes at the programmatic level is the `openat(2)` function. Once a file descriptor has been obtained for an attribute file by an `openat()` call, all normal file system semantics apply. There is no attempt to place special semantics on `read(2)`, `write(2)`, `ftruncate(3C)`, or other functions when applied to attribute file descriptors relative to "normal" file descriptors.

The set of existing attributes can be browsed by calling `openat()` with "." as the file name and the `O_XATTR` flag set, resulting in a file descriptor for the attribute directory. The list of attributes is obtained by calls to `getdents(2)` on the returned file descriptor. If the target file did not previously have any attributes associated with it, an empty top-level attribute directory

is created for the file and subsequent `getdents()` calls will return only "." and ". .". While the owner of the parent file owns the extended attribute directory, it is not charged against its quota if the directory is empty. Attribute files themselves, however, are charged against the user quota as any other regular file.

Additional system calls have been provided as convenience functions, including `faccessat(2)`, `fchownat(2)`, `fstatat(2)`, `futimesat(2)`, `renameat(2)`, `unlinkat(2)`. These new functions, along with `openat()`, provide a mechanism to access files relative to an arbitrary point in the file system, rather than only the current working directory. This mechanism is particularly useful in situations when a file descriptor is available with no path. The `openat()` function, in particular, can be used in many contexts where `chdir()` or `fchdir()` is currently required. See `chdir(2)`.

Open a file relative to a file descriptor

`int openat (int` *fd*`, const char *`*path*`, int` *oflag* `[, mode_t` *mode*`])`

The `openat(2)` function behaves exactly as `open(2)` except when given a relative path. Where `open()` resolves a relative path from the current working directory, `openat()` resolves the path based on the vnode indicated by the supplied file descriptor. When *oflag* is `O_XATTR`, `openat()` interprets the *path* argument as an extended attribute reference. The following code fragment uses `openat()` to examine the attributes of some already opened file:

```
dfd = openat(fd, ".", O_RDONLY|O_XATTR);
(void)getdents(dfd, buf, nbytes);
```

If `openat()` is passed the special value `AT_FDCWD` as its first (*fd*) argument, its behavior is identical to `open()` and the relative path arguments are interpreted relative to the current working directory. If the `O_XATTR` flag is provided to `openat()` or to `open()`, the supplied path is interpreted as a reference to an extended attribute on the current working directory.

Unlink a file relative to a directory file descriptor

`int unlinkat (int` *dirfd*`, const char *path`*flag*`, int flag`*flag*`)`

The `unlinkat(2)` function deletes an entry from a directory. The *path* argument indicates the name of the entry to remove. If *path* an absolute path, the *dirfd* argument is ignored. If it is a relative path, it is interpreted relative to the directory indicated by the *dirfd* argument. If *dirfd* does not refer to a valid directory, the function returns `ENOTDIR`. If the special value `AT_FDCWD` is specified for *dirfd*, a relative path argument is resolved relative to the current working directory. If the *flag* argument is 0, all other semantics of this function are equivalent to `unlink(2)`. If *flag* is set to `AT_REMOVEDIR`, all other semantics of this function are equivalent to `rmdir(2)`.

Rename a file relative to directories

`int renameat (int` *fromfd*`, const char *`*old*`, int` *tofd*`, const char *`*new*`)`

The renameat(2) function renames an entry in a directory, possibly moving the entry into a different directory. The *old* argument indicates the name of the entry to rename. If this argument is a relative path, it is interpreted relative to the directory indicated by the *fd* argument. If it is an absolute path, the *fromfd* argument is ignored. The *new* argument indicates the new name for the entry. If this argument is a relative path, it is interpreted relative to the directory indicated by the *tofd* argument. If it is an absolute path, the *tofd* argument is ignored.

In the relative path cases, if the directory file descriptor arguments do not refer to a valid directory, the function returns ENOTDIR. All other semantics of this function are equivalent to rename(2).

If a special value AT_FDCWD is specified for either the *fromfd* or *tofd* arguments, their associated path arguments (*old* and *new*) are interpreted relative to the current working directory if they are not specified as absolute paths. Any attempt to use renameat() to move a file that is not an extended attribute into an extended attribute directory (so that it becomes an extended attribute) will fail. The same is true for an attempt to move a file that is an extended attribute into a directory that is not an extended attribute directory.

Obtain information about a file

```
int fstatat (int fd, const char *path, struct stat* buf, int flag)
```

The fstatat(2) function obtains information about a file. If the *path* argument is relative, it is resolved relative to the *fd* argument file descriptor, otherwise the *fd* argument is ignored. If the *fd* argument is a special value AT_FDCWD the path is resolved relative to the current working directory. If the *path* argument is a null pointer, the function returns information about the file referenced by the *fd* argument. In all other relative path cases, if the *fd* argument does not refer to a valid directory, the function returns ENOTDIR. If AT_SYMLINK_NOFOLLOW is set in the *flag* argument, the function will not automatically traverse a symbolic link at the position of the path. If _AT_TRIGGER is set in the *flag* argument and the vnode is a trigger mount point, the mount is performed and the function returns the attributes of the root of the mounted filesystem. The fstatat() function is a multipurpose function that can be used in place of stat(), lstat(), or fstat(). See stat(2)

The function call stat(*path*, *buf*) is identical to fstatat(AT_FDCWD, *path*, *buf*, 0).

The function call lstat(*path*, *buf*) is identical to fstatat(AT_FDCWD, *path*, *buf*, AT_SYMLINK_NOFOLLOW)

The function call fstat(*fildes*, *buf*) is identical to fstatat(*fildes*, NULL, *buf*, 0).

Set owner and group ID

```
int fchownat (int fd, const char *path, uid_t owner, gid_t group, \
          int flag)
```

The fchownat(2) function sets the owner ID and group ID for a file. If the *path* argument is relative, it is resolved relative to the *fd* argument file descriptor, otherwise the *fd* argument is ignored. If the *fd* argument is a special value AT_FDCWD the path is resolved relative to the current working directory. If the path argument is a null pointer, the function sets the owner and group ID of the file referenced by the *fd* argument. In all other relative path cases, if the *fd* argument does not refer to a valid directory, the function returns ENOTDIR. If the *flag* argument is set to AT_SYMLINK_NOFOLLOW, the function will not automatically traverse a symbolic link at the position of the path. The fchownat() function is a multi-purpose function that can be used in place of chown(), lchown(), or fchown(). See chown(2).

The function call chown(*path*, *owner*, *group*) is equivalent to fchownat(AT_FDCWD, *path*, *owner*, *group*, 0).

The function call lchown(*path*, *owner*, *group*) is equivalent to fchownat(AT_FDCWD, *path*, *owner*, *group*, AT_SYMLINK_NOFOLLOW).

Set file access and modification times

```
int futimesat (int fd, const char *path, const struct timeval \
              times[2])
```

The futimesat(2) function sets the access and modification times for a file. If the *path* argument is relative, it is resolved relative to the *fd* argument file descriptor; otherwise the *fd* argument is ignored. If the *fd* argument is the special value AT_FDCWD, the path is resolved relative to the current working directory. If the *path* argument is a null pointer, the function sets the access and modification times of the file referenced by the *fd* argument. In all other relative path cases, if the *fd* argument does not refer to a valid directory, the function returns ENOTDIR. The futimesat() function can be used in place of utimes(2).

The function call utimes(*path*, *times*) is equivalent to futimesat(AT_FDCWD, *path*, *times*).

Determine accessibility of a file

```
int faccessat(int fd, const char *path, int amode, int flag);
```

The faccessat() function checks the file named by the pathname pointed to by the *path* argument for accessibility according to the bit pattern contained in *amode*, using the real user ID in place of the effective user ID and the real group ID in place of the effective group ID. This allows a setuid process to verify that the user running it would have had permission to access this file.

If *path* specifies a relative path, the file whose accessibility is to be determined is located relative to the directory associated with the file descriptor *fd* instead of the current working directory. If *path* specifies an absolute path, the *fd* argument is ignored.

If `faccessat()` is passed in the *fd* parameter the special value `AT_FDCWD`, defined in `<fcntl.h>`, the current working directory is used and the behavior is identical to a call to `access(2)`.

New `pathconf()` functionality

```
long int pathconf(const char *path, int name)
```

Two variables have been added to `pathconf(2)` to provide enhanced support for extended attribute manipulation. The `XATTR_ENABLED` variable allows an application to determine if attribute support is currently enabled for the file in question. The `XATTR_EXISTS` variable allows an application to determine whether there are any extended attributes associated with the supplied path.

Open/Create an attribute file

```
int attropen (const char *path, const char *attrpath, int oflag \
        [, mode_t mode])
```

The `attropen(3C)` function returns a file descriptor for the named attribute, *attrpath*, of the file indicated by *path*. The *oflag* and *mode* arguments are identical to the `open(2)` arguments and are applied to the open operation on the attribute file (for example, using the `O_CREAT` flag creates a new attribute). Once opened, all normal file system operations can be used on the attribute file descriptor. The `attropen()` function is a convenience function and is equivalent to the following sequence of operations:

```
fd = open (path, O_RDONLY);
attrfd = openat(fd, attrpath, oflag|O_XATTR, mode);
close(fd);
```

The set of existing attributes can be browsed by calling `attropen()` with "." as the attribute name. The list of attributes is obtained by calling `getdents(2)` (or `fdopendir(3C)` followed by `readdir(3C)`, see below) on the returned file descriptor.

Convert an open file descriptor for a directory into a directory descriptor

```
DIR * fdopendir (const int fd)
```

The `fdopendir(3C)` function promotes a file descriptor for a directory to a directory pointer suitable for use with the `readdir(3C)` function. The originating file descriptor should not be used again following the call to `fdopendir()`. The directory pointer should be closed with a call to `closedir(3C)`. If the provided file descriptor does not reference a directory, the function returns `ENOTDIR`. This function is useful in circumstances where the only available handle on a directory is a file descriptor. See `attropen(3C)` and `openat(2)`.

Using the API

The following examples demonstrate how the API might be used to perform basic operations on extended attributes:

例 1 List extended attributes on a file.

```
attrdirfd = attropen("test", ".", O_RDONLY);
dirp = fdopendir(attrdirfd);
while (dp = readdir(dirp)) {
...
```

例 2 Open an extended attribute.

```
attrfd = attropen("test", dp->d_name, O_RDONLY);
```

or

```
attrfd = openat(attrdirfd, dp->d_name, O_RDONLY);
```

例 3 Read from an extended attribute.

```
while (read(attrfd, buf, 512) > 0) {
...
```

例 4 Create an extended attribute.

```
newfd = attropen("test", "attr", O_CREAT|O_RDWR);
```

or

```
newfd = openat(attrdirfd, "attr", O_CREAT|O_RDWR);
```

例 5 Write to an extended attribute.

```
count = write(newfd, buf, length);
```

例 6 Delete an extended attribute.

```
error = unlinkat(attrdirfd, "attr");
```

Applications intending to access the interfaces defined here as well as the POSIX and X/Open specification-conforming interfaces should define the macro `_ATFILE_SOURCE` to be 1 and set whichever feature test macros are appropriate to obtain the desired environment. See `standards(5)`.

Extended Archive Formats

As noted above in the description of command utilities modified to provide support for extended attributes, the archive formats for `tar(1)` and `cpio(1)` have been extended to provide support for archiving extended attributes. This section describes the specifics of the archive format extensions.

Extended tar format

The `tar` archive is made up of a series of 512 byte blocks. Each archived file is represented by a header block and zero or more data blocks containing the file contents. The header block is structured as shown in the following table.

Field Name	Length (in Octets)	Description
Name	100	File name string
Mode	8	12 file mode bits
Uid	8	User ID of file owner
Gid	8	Group ID of file owner
Size	12	Size of file
Mtime	12	File modification time
Chksum	8	File contents checksum
Typeflag	1	File type flag
Linkname	100	Link target name if file linked
Magic	6	“ustar”
Version	2	“00”
Uname	32	User name of file owner
Gname	32	Group name of file owner
Devmajor	8	Major device ID if special file
Devminor	8	Minor device ID if special file
Prefix	155	Path prefix string for file

The extended attribute project extends the above header format by defining a new header type (for the `Typeflag` field). The type 'E' is defined to be used for all extended attribute files. Attribute files are stored in the `tar` archive as a sequence of two `<header ,data>` pairs. The first file contains the data necessary to locate and name the extended attribute in the file system. The second file contains the actual attribute file data. Both files use an 'E' type header. The prefix and name fields in extended attribute headers are ignored, though they should be set to meaningful values for the benefit of archivers that do not process these headers. Solaris archivers set the prefix field to “`/dev/null`” to prevent archivers that do not understand the type 'E' header from trying to restore extended attribute files in inappropriate places.

Extended cpio format

The `cpio` archive format is octet-oriented rather than block-oriented. Each file entry in the archive includes a header that describes the file, followed by the file name, followed by the contents of the file. These data are arranged as described in the following table.

Field Name	Length (in Octets)	Description
c_magic	6	70707
c_dev	6	First half of unique file ID
c_ino	6	Second half of unique file ID
c_mode	6	File mode bits
c_uid	6	User ID of file owner
c_gid	6	Group ID of file owner
c_nlink	6	Number of links referencing file
c_rdev	6	Information for special files
c_mtime	11	Modification time of file
c_namesize	6	Length of file pathname
c_filesize	11	Length of file content
c_name	c_namesize	File pathname
c_filedata	c_filesize	File content

The basic archive file structure is not changed for extended attributes. The file type bits stored in the `c_mode` field for an attribute file are set to `0xB000`. As with the `tar` archive format, extended attributes are stored in `cpio` archives as two consecutive file entries. The first file describes the location/name for the extended attribute. The second file contains the actual attribute file content. The `c_name` field in extended attribute headers is ignored, though it should be set to a meaningful value for the benefit of archivers that do not process these headers. Solaris archivers start the pathname with "`/dev/null/`"to prevent archivers that do not understand the type 'E' header from trying to restore extended attribute files in inappropriate places.

Attribute identification data format

Both the `tar` and `cpio` archive formats can contain the special files described above, always paired with the extended attribute data record, for identifying the precise location of the extended attribute. These special data files are necessary because there is no simple naming mechanism for extended attribute files. Extended attributes are not visible in the file system name space. The extended attribute name space must be "tunneled into" using the `openat()` function. The attribute identification data must support not only the flat naming structure for extended attributes, but also the possibility of future extensions allowing for attribute directory hierarchies and recursive attributes. The data file is therefore composed of a sequence of records. It begins with a fixed length header describing the content. The following table describes the format of this data file.

Field Name	Length (in Octets)	Description
h_version	7	Name file version
h_size	10	Length of data file
h_component_len	10	Total length of all path segments
h_link_comp_len	10	Total length of all link segments
path	h_component_len	Complex path
link_path	h_link_comp_len	Complex link path

As demonstrated above, the header is followed by a record describing the "path" to the attribute file. This path is composed of two or more path segments separated by a null character. Each segment describes a path rooted at the hidden extended attribute directory of the leaf file of the previous segment, making it possible to name attributes on attributes. The first segment is always the path to the parent file that roots the entire sequence in the normal name space. The following table describes the format of each segment.

Field Name	Length (in Octets)	Description
h_namesz	7	Length of segment path
h_typeflag	1	Actual file type of attribute file
h_names	h_namesz	Parent path + segment path

If the attribute file is linked to another file, the path record is followed by a second record describing the location of the referencing file. The structure of this record is identical to the record described above.

関連項目

cp(1), cpio(1), find(1), ls(1), mv(1), pax(1), runat(1), tar(1), du(1), fsck(1M), access(2), chown(2), link(2), open(2), pathconf(2), rename(2), stat(2), unlink(2), utimes(2), attropen(3C), standards(5)

名前

grub – Oracle Solaris の GRand Unified Bootloader 2 ソフトウェア

機能説明

Oracle Solaris オペレーティングシステムの現行リリースには、GRUB (GRand Unified Bootloader) 2 ソフトウェアが付属しています。GRUB は、Free Software Foundation によって開発およびサポートされています。

`www.gnu.org` からアクセス可能な GRUB マニュアルの「Overview」では、GRUB を次のように説明しています。

> Briefly, a boot loader is the first software program that runs when a computer starts. It is responsible for loading and transferring control to an operating system kernel software (such as Linux or GNU Mach). The kernel, in turn, initializes the rest of the operating system (for example, a GNU [Ed. note: or Solaris] system).
>
> GNU GRUB は、幅広い無料および商用のオペレーティングシステムをチェーンロードせずにロードできる非常に強力なブートローダーです。GRUB は、複雑なパーソナルコンピュータのブートに対処するように設計されています。今後、ほかのプラットフォームへの移植が検討される可能性もありますが、プログラムとこのドキュメントの両方は該当するコンピュータプラットフォームに緊密に結合されています。注: Oracle Solaris オペレーティングシステムには GRUB が移植されています。
>
> One of the important features in GRUB is flexibility; GRUB understands filesystems and kernel executable formats, so you can load an arbitrary operating system the way you like, without recording the physical position of your kernel on the disk. Thus you can load the kernel just by specifying its file name and the drive and partition where the kernel resides.

GRUB がサポートされるのは、Solaris マシンのうち、x86 プラットフォームのみです。Solaris に付属する GRUB ソフトウェアには、オープンソースディストリビューションには含まれていない、次のユーティリティーが追加されています:

`bootadm(1M)` ブートアーカイブの管理と GRUB メニューの変更を可能にします。

この Solaris 固有ユーティリティー以外の GRUB ソフトウェアについては、GRUB ドキュメント内で説明されています。その PDF 版は Oracle の Web サイトから入手可能となっています。

`bootadm(1M)` には、GRUB メニューから変更できるインタフェースが備わっているため、ほとんどの管理者は、GRUB ユーティリティー (BIOS ファームウェアの場合は `/usr/lib/grub2/bios`、64 ビットの UEFI ファームウェアの場合は `/usr/lib/grub2/uefi64` にあります) を手動で実行する必要はありません。

管理者は GRUB 構成ファイル (`grub.cfg`) を手動で編集しないでください。このファイルは、システムのルートプール (つまり、`/rpool/boot/grub`) の最上位 ZFS データセットの `/boot/grub` サブディレクトリにあります。このファイルは、管理者が `bootadm(1M)` を使用してブート構成を変更すると自動的に再生成されま

す。GRUB メニューエントリを直接作成する上級管理者は、grub.cfg ファイルと同じディレクトリに格納されている補足の GRUB 構成ファイル named custom.cfg を編集する必要があります。新たにインストールされたシステムには、このファイルは存在しないため、管理者が作成する必要があることに注意してください。

関連項目

boot(1M), bootadm(1M), installgrub(1M)

『Oracle Solaris 11.1 システムのインストール』

『Oracle Solaris の管理: 一般的なタスク』

http://www.gnu.org/software/grub

注意事項

以前のバージョンの Solaris では、GRUB ブートローダーをインストールするための installgrub(1M) コマンドの使用法についてドキュメント化していました。このコマンドは非推奨になり、利便性、障害回復、および GRUB レガシーがシステムブートローダーとなっている Solaris ブート環境にダウングレードするためにだけ存在します。詳細については、Oracle Solaris のブートおよびシャットダウンのドキュメントを参照してください。

名前

gss_auth_rules – overview of GSS authorization

機能説明

The establishment of the veracity of a user's credentials requires both authentication (Is this an authentic user?) and authorization (Is this authentic user, in fact, authorized?).

When a user makes use of Generic Security Services (GSS) versions of the `ftp` or `ssh` clients to connect to a server, the user is not necessarily authorized, even if his claimed GSS identity is authenticated, Authentication merely establishes that the user is who he says he is to the GSS mechanism's authentication system. Authorization is then required: it determines whether the GSS identity is permitted to access the specified Solaris user account.

The GSS authorization rules are as follows:

- If the mechanism of the connection has a set of authorization rules, then use those rules. For example, if the mechanism is Kerberos, then use the `krb5_auth_rules(5)`, so that authorization is consistent between raw Kerberos applications and GSS/Kerberos applications.
- If the mechanism of the connection does not have a set of authorization rules, then authorization is successful if the remote user's `gssname` matches the local user's `gssname` exactly, as compared by `gss_compare_name(3GSS)`.

ファイル

`/etc/passwd` System account file. This information may also be in a directory service. See `passwd(4)`.

属性

See `attributes(5)` for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

`ftp(1)`, `ssh(1)`, `gsscred(1M)`, `gss_compare_name(3GSS)`, `passwd(4)`, `attributes(5)`, `krb5_auth_rules(5)`

名前

hal – ハードウェア抽象化レイヤーの概要

機能説明

ハードウェア抽象化レイヤー (HAL) は、システムに接続されている各種ハードウェアのビューを提供します。このビューは、ハードウェア構成に変化があると、ホットプラグなどのメカニズムを通じて動的に更新されます。HAL は、ハードウェアの断片をデバイスオブジェクトとして表します。デバイスオブジェクトは、一意の識別子によって識別され、デバイスプロパティーと呼ばれる一連のキーと値のペアを保持します。プロパティーには、実際のハードウェアから取得されるもの、デバイス情報ファイル (`.fdi` ファイル) からマージされるもの、実際のデバイス構成に関連しているものがあります。

HAL は、D-Bus を通して使いやすい API を提供します。D-Bus はさまざまな機能を持つ IPC フレームワークであり、特に、アプリケーション間のやりとりを可能にするシステム全体のメッセージバスを提供します。具体的には、D-Bus は非同期の通知を提供するため、デバイスの追加と削除、およびデバイスのプロパティーの変更が行われたときに、HAL はメッセージバス上のほかの相手に通知できます。

Solaris オペレーティングシステムでは、HAL は `hald(1M)` デーモン、およびデバイスの追加と削除およびそのプロパティーの変更を行う一連のユーティリティーでサポートされています。

関連項目

`hald(1M)`, `hal-device(1M)`, `hal-fdi-validate(1M)`, `hal-find(1M)`, `hal-find-by-capability(1M)`, `hal-find-by-property(1M)`, `hal-get-property(1M)`, `hal-set-property(1M)`, `fdi(4)`

http://freedesktop.org。

名前 iconv_1250 – code set conversion tables for MS 1250 (Windows Latin 2)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
MS 1250	win2	ISO 8859-2	iso2	ISO Latin 2
MS 1250	win2	MS 852	dos2	MS-DOS Latin 2
MS 1250	win2	Mazovia	maz	Mazovia
MS 1250	win2	DHN	dhn	Dom Handlowy Nauki

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

MS 1250 to ISO 8859-2 For the conversion of MS 1250 to ISO 8859-2, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1250	ISO 8859-2	MS 1250	ISO 8859-2
24-211	40	235	273
212	251	236	276
213	40	237	274
214	246	241	267
215	253	245	241
216	256	246-267	40
217	254	271	261
221-231	40	273	40
232	271	274	245
233	40	276	265
234	266	247	365

MS 1250 to MS 852 For the conversion of MS 1250 to MS 852, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1250	MS 852	MS 1250	MS 852
200-211	40	311	220
212	346	312	250
213	40	313	323
214	227	314	267
215	233	315	326
216	246	316	327
217	215	317	322
220-231	40	320	321
232	347	321	343
233	40	322	325
234	230	323	340
235	234	324	342
236	247	325	212
237	253	326	231
240	377	327	236
241	363	330	374
242	364	331	336
243	235	332	351
244	317	333	353
245	244	334	232
246	40	335	355
247	365	336	335
250	371	337	341
251	40	340	352
252	270	341	240
253	256	342	203
254	252	343	307

Conversions Performed			
MS 1250	MS 852	MS 1250	MS 852
255	360	344	204
256	40	345	222
257	275	346	206
260	370	347	207
261	40	350	237
262	362	351	202
263	210	352	251
264	357	353	211
265-267	40	354	330
270	367	355	241
271	245	356	214
272	255	357	324
273	257	360	320
274	225	361	344
275	361	362	345
276	226	363	242
277	276	364	223
300	350	365	213
301	265	366	224
302	266	367	366
303	306	370	375
304	216	371	205
305	221	372	243
306	217	374	201
307	200	375	354
310	254	376	356

MS 1250 to Mazovia

For the conversion of MS 1250 to Mazovia, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1250	**Mazovia**	**MS 1250**	**Mazovia**
200-213	40	310-311	40
214	230	312	220
215-216	40	313-320	40
217	240	321	245
220-233	40	322	40
234	236	323	243
235-236	40	324-325	40
237	246	326	231
240	377	327-333	40
241-242	40	334	232
243	234	335-336	40
244	40	337	341
245	217	340-341	40
246-252	40	342	203
253	256	343	40
254	252	344	204
255-256	40	345	40
257	241	346	215
260	370	347	207
261	361	350	40
262	40	351	202
263	222	352	221
264	40	353	211
265	346	354-355	40
266	40	356	214

Conversions Performed			
MS 1250	Mazovia	MS 1250	Mazovia
267	372	357-360	40
270	40	361	244
271	206	362	40
272	40	363	242
273	257	364	223
274-276	40	365	40
277	247	366	224
300-303	40	367	366
304	216	370-373	40
305	40	374	201
306	225	375-376	40
307	200		

MS 1250 to DHN

For the conversion of MS 1250 to DHN, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1250	DHN	MS 1250	DHN
200-213	40	306	201
214	206	307-311	40
215-216	40	312	202
217	207	313-320	40
220-233	40	321	204
234	217	322	40
235-236	40	323	205
237	220	324-325	40
240	377	326	231
241-242	40	327-333	40
243	203	334	232

Conversions Performed			
MS 1250	DHN	MS 1250	DHN
244	40	335-336	40
245	200	337	341
246-252	40	340	40
253	256	341	240
254	252	342-345	40
255-256	40	346	212
257	210	347-351	40
260	370	352	213
261	361	353-354	40
262	40	355	241
263	214	356-360	40
264	40	361	215
265	346	362	40
266	40	363	216
267	372	364	223
270	40	365	40
271	211	366	224
272	40	367	366
273	257	370-371	40
274-276	40	372	243
277	221	373-376	40
300-305	40		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

iconv(1), iconv(3C), iconv(5)

名前 iconv_1251 – code set conversion tables for MS 1251 (Windows Cyrillic)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
MS 1251	win5	ISO 8859-5	iso5	ISO 8859-5 Cyrillic
MS 1251	win5	KOI8-R	koi8	KOI8-R
MS 1251	win5	PC Cyrillic	alt	Alternative PC Cyrillic
MS 1251	win5	Mac Cyrillic	mac	Macintosh Cyrillic

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

MS 1251 to ISO 8859-5 For the conversion of MS 1251 to ISO 8859-5, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1251	ISO 8859-5	MS 1251	ISO 8859-5
24	4	310	270
200	242	311	271
201	243	312	272
202	40	313	273
203	363	314	274
204-207	40	315	275
210	255	316	276
211	40	317	277
212	251	320	300
213	40	321	301
214	252	322	302
215	254	323	303
216	253	324	304
217	257	325	305

Conversions Performed			
MS 1251	ISO 8859-5	MS 1251	ISO 8859-5
220	362	326	306
221-227	40	327	307
230	255	330	310
231	40	331	311
232	371	332	312
233	40	333	313
234	372	334	314
235	374	335	315
236	373	336	316
237	377	337	317
241	256	340	320
242	376	341	321
243	250	342	322
244-247	40	343	323
250	241	344	324
251	40	345	325
252	244	346	326
253-254	40	347	327
255	55	350	330
256	40	351	331
257	247	352	332
260-261	40	353	333
262	246	354	334
263	366	355	335
264-267	40	356	336
270	361	357	337
271	360	360	340

Conversions Performed			
MS 1251	ISO 8859-5	MS 1251	ISO 8859-5
272	364	361	341
273	40	362	342
274	370	363	343
275	245	364	344
276	365	365	345
277	367	366	346
300	260	367	347
301	261	370	350
302	262	371	351
303	263	372	352
304	264	373	353
305	265	374	354
306	266	375	355
307	267	376	356

MS 1251 to KOI8-R

For the conversion of MS 1251 to KOI8-R , all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1251	KOI8-R	MS 1251	KOI8-R
24	4	310	351
200	261	311	352
201	262	312	353
202	40	313	354
203	242	314	355
204-207	40	315	356
210	255	316	357
211	40	317	360
212	271	320	362

Conversions Performed			
MS 1251	KOI8-R	MS 1251	KOI8-R
213	40	321	363
214	272	322	364
215	274	323	365
216	273	324	346
217	277	325	350
220	241	326	343
221-227	40	327	376
230	255	330	373
231	40	331	375
232	251	332	377
233	40	333	371
234	252	334	370
235	254	335	374
236	253	336	340
237	257	337	361
241	276	340	301
242	256	341	302
243	270	342	327
244-247	40	343	307
250	263	344	304
251	40	345	305
252	264	346	326
253-254	40	347	332
255	55	350	311
256	40	351	312
257	267	352	313
260-261	40	353	314

Conversions Performed			
MS 1251	KOI8-R	MS 1251	KOI8-R
262	266	354	315
263	246	355	316
264-267	40	356	317
270	243	357	320
271	260	360	322
272	244	361	323
273	40	362	324
274	250	363	325
275	265	364	306
276	245	365	310
277	247	366	303
300	341	367	336
301	342	370	333
302	367	371	335
303	347	372	337
304	344	373	331
305	345	374	330
306	366	375	334
307	372	376	300

MS 1251 to PC Cyrillic

For the conversion of MS 1251 to PC Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1251	PC Cyrillic	MS 1251	PC Cyrillic
24	4	332	232
200-207	40	333	233
210	260	334	234
211-227	40	335	235

Conversions Performed			
MS 1251	PC Cyrillic	MS 1251	PC Cyrillic
230	260	336	236
231-247	40	337	237
250	360	340	240
251-254	40	341	241
255	55	342	242
256-267	40	343	243
270	361	344	244
271-277	40	345	245
300	200	346	246
301	201	347	247
302	202	350	250
303	203	351	251
304	204	352	252
305	205	353	253
306	206	354	254
307	207	355	255
310	210	356	256
311	211	357	257
312	212	360	340
313	213	361	341
314	214	362	342
315	215	363	343
316	216	364	344
317	217	365	345
320	220	366	346
321	221	367	347
322	222	370	350

Conversions Performed			
MS 1251	PC Cyrillic	MS 1251	PC Cyrillic
323	223	371	351
324	224	372	352
325	225	373	353
326	226	374	354
327	227	375	355
330	230	376	356
331	231		

MS 1251 to Mac Cyrillic

For the conversion of MS 1251 to Mac Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 1251	Mac Cyrillic	MS 1251	Mac Cyrillic
24	4	260	241
200	253	262	247
201	256	263	264
202	40	264	266
203	257	266	246
204	327	267	245
205	311	270	336
206	240	271	334
207-211	40	272	271
212	274	273	310
213	40	274	300
214	276	275	301
215	315	276	317
216	40	277	273
217	332	300	200
220	254	301	201

Conversions Performed			
MS 1251	Mac Cyrillic	MS 1251	Mac Cyrillic
221	324	302	202
222	325	303	203
223	322	304	204
224	323	305	205
225	40	306	206
226	320	307	207
227	321	310	210
230	40	311	211
231	252	312	212
232	275	313	213
233	40	314	214
234	277	315	215
235	316	316	216
236	40	317	217
237	333	320	220
240	312	321	221
241	330	322	222
242	331	323	223
243	267	324	224
244	377	325	225
245	242	326	226
246	40	327	227
247	244	330	230
250	335	331	231
252	270	332	232
253	307	333	233
254	302	334	234

Conversions Performed			
MS 1251	Mac Cyrillic	MS 1251	Mac Cyrillic
255	55	335	235
256	250	336	236
257	272	337	237
355	316		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

iconv(1), iconv(3C), iconv(5)

名前　　iconv – コードセット変換テーブル

機能説明　　以下に示すようなコードセット変換が可能です。

Code Set Conversions Supported

Code	Symbol	Target Code	Symbol	Target Output
ISO 646	646	ISO 8859-1	8859	US ASCII
ISO 646de	646de	ISO 8859-1	8859	German
ISO 646da	646da	ISO 8859-1	8859	Danish
ISO 646en	646en	ISO 8859-1	8859	English ASCII
ISO 646es	646es	ISO 8859-1	8859	Spanish
ISO 646fr	646fr	ISO 8859-1	8859	French
ISO 646it	646it	ISO 8859-1	8859	Italian
ISO 646sv	646sv	ISO 8859-1	8859	Swedish
ISO 8859-1	8859	ISO 646	646	7 bit ASCII
ISO 8859-1	8859	ISO 646de	646de	German
ISO 8859-1	8859	ISO 646da	646da	Danish
ISO 8859-1	8859	ISO 646en	646en	English ASCII
ISO 8859-1	8859	ISO 646es	646es	Spanish
ISO 8859-1	8859	ISO 646fr	646fr	French
ISO 8859-1	8859	ISO 646it	646it	Italian
ISO 8859-1	8859	ISO 646sv	646sv	Swedish
ISO 8859-16	iso16	ISO 8859-2	iso2	ISO Latin 2
ISO 8859-2	iso2	ISO 8859-16	iso16	ISO Latin 10
ISO 8859-16	iso16	IBM 850	ibm850	IBM 850 code page
ISO 8859-16	iso16	IBM 870	ibm870	IBM 870 code page
ISO 8859-2	iso2	MS 1250	win2	Windows Latin 2
ISO 8859-2	iso2	MS 852	dos2	MS-DOS Latin 2
ISO 8859-2	iso2	Mazovia	maz	Mazovia
IBM 850	ibm850	ISO 8859-16	iso16	ISO Latin 10
IBM 870	ibm870	ISO 8859-16	iso16	ISO Latin 10
MS 1250	win2	DHN	dhn	Dom Handlowy Nauki
MS 852	dos2	ISO 8859-2	iso2	ISO Latin 2
MS 852	dos2	MS 1250	win2	Windows Latin 2
MS 852	dos2	Mazovia	maz	Mazovia
MS 852	dos2	DHN	dhn	Dom Handlowy Nauki
Mazovia	maz	ISO 8859-2	iso2	ISO Latin 2
Mazovia	maz	MS 1250	win2	Windows Latin 2
Mazovia	maz	MS 852	dos2	MS-DOS Latin 2
Mazovia	maz	DHN	dhn	Dom Handlowy Nauki
DHN	dhn	ISO 8859-2	iso2	ISO Latin 2
DHN	dhn	MS 1250	win2	Windows Latin 2
DHN	dhn	MS 852	dos2	MS-DOS Latin 2
DHN	dhn	Mazovia	maz	Mazovia
ISO 8859-5	iso5	KOI8-R	koi8	KOI8-R
ISO 8859-5	iso5	PC Cyrillic	alt	Alternative PC Cyrillic

```
ISO 8859-5      iso5      MS 1251         win5      Windows Cyrillic
ISO 8859-5      iso5      Mac Cyrillic    mac       Macintosh Cyrillic
KOI8-R          koi8      ISO 8859-5      iso5      ISO 8859-5 Cyrillic
KOI8-R          koi8      PC Cyrillic     alt       Alternative PC Cyrillic
KOI8-R          koi8      MS 1251         win5      Windows Cyrillic
KOI8-R          koi8      Mac Cyrillic    mac       Macintosh Cyrillic
PC Cyrillic     alt       ISO 8859-5      iso5      ISO 8859-5 Cyrillic
PC Cyrillic     alt       KOI8-R          koi8      KOI8-R
PC Cyrillic     alt       MS 1251         win5      Windows Cyrillic
PC Cyrillic     alt       Mac Cyrillic    mac       Macintosh Cyrillic
MS 1251         win5      ISO 8859-5      iso5      ISO 8859-5 Cyrillic
MS 1251         win5      KOI8-R          koi8      KOI8-R
MS 1251         win5      PC Cyrillic     alt       Alternative PC Cyrillic
MS 1251         win5      Mac Cyrillic    mac       Macintosh Cyrillic
Mac Cyrillic    mac       ISO 8859-5      iso5      ISO 8859-5 Cyrillic
Mac Cyrillic    mac       KOI8-R          koi8      KOI8-R
Mac Cyrillic    mac       PC Cyrillic     alt       Alternative PC Cyrillic
Mac Cyrillic    mac       MS 1251         win5      Windows Cyrillic
```

コード変換

変換処理は、次の「コード変換の索引」に記述されたマニュアルページに含まれる表に従って行われます。

コード変換の索引		
コード	出力側コード	参照マニュアルページ
ISO 646	ISO 8859-1	iconv_646 (5)
ISO 646de	ISO 8859-1	
ISO 646da	ISO 8859-1	
ISO 646en	ISO 8859-1	
ISO 646es	ISO 8859-1	
ISO 646fr	ISO 8859-1	
ISO 646it	ISO 8859-1	
ISO 646sv	ISO 8859-1	
ISO 8859-1	ISO 646	iconv_8859-1 (5)
ISO 8859-1	ISO 646de	
ISO 8859-1	ISO 646da	
ISO 8859-1	ISO 646en	
ISO 8859-1	ISO 646es	

コード変換の索引		
ISO 8859-1	ISO 646fr	
ISO 8859-1	ISO 646it	
ISO 8859-1	ISO 646sv	
ISO 8859-2	MS 1250	iconv_8859-2 (5)
ISO 8859-2	MS 852	
ISO 8859-2	Mazovia	
ISO 8859-2	DHN	
MS 1250	ISO 8859-2	iconv_1250 (5)
MS 1250	MS 852	
MS 1250	Mazovia	
MS 1250	DHN	
MS 852	ISO 8859-2	iconv_852 (5)
MS 852	MS 1250	
MS 852	Mazovia	
MS 852	DHN	
Mazovia	ISO 8859-2	iconv_maz (5)
Mazovia	MS 1250	
Mazovia	MS 852	
Mazovia	DHN	

コード変換の索引		
コード	出力側コード	参照マニュアルページ
DHN	ISO 8859-2	iconv_dhn (5)
DHN	MS 1250	
DHN	MS 852	
DHN	Mazovia	
ISO 8859-5	KOI8-R	iconv_8859-5 (5)
ISO 8859-5	PC Cyrillic	

コード変換の索引		
ISO 8859-5	MS 1251	
ISO 8859-5	Mac Cyrillic	
KOI8-R	ISO 8859-5	iconv_koi8-r (5)
KOI8-R	PC Cyrillic	
KOI8-R	MS 1251	
KOI8-R	Mac Cyrillic	
PC Cyrillic	ISO 8859-5	iconv_pc_cyr (5)
PC Cyrillic	KOI8-R	
PC Cyrillic	MS 1251	
PC Cyrillic	Mac Cyrillic	
MS 1251	ISO 8859-5	iconv_1251 (5)
MS 1251	KOI8-R	
MS 1251	PC Cyrillic	
MS 1251	Mac Cyrillic	
Mac Cyrillic	ISO 8859-5	iconv_mac_cyr (5)
Mac Cyrillic	KOI8-R	
Mac Cyrillic	PC Cyrillic	
Mac Cyrillic	MS 1251	

ファイル

`/usr/lib/iconv/*.so`
　変換モジュール

`/usr/lib/iconv/*.t`
　変換テーブル

`/usr/lib/iconv/geniconvtbl/binarytables/*.bt`
　バイナリ形式の変換テーブル

`/usr/lib/iconv/iconv_data`
　変換テーブルがサポートしている変換のリスト

関連項目

iconv(1), iconv(3C), iconv_1250(5), iconv_1251(5), iconv_646(5), iconv_852(5), iconv_8859-1(5), iconv_8859-2(5), iconv_8859-5(5), iconv_dhn(5), iconv_koi8-r(5), iconv_mac_cyr(5), iconv_maz(5), iconv_pc_cyr(5), iconv_unicode(5)

名前 iconv_646 – code set conversion tables for ISO 646

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	**Symbol**	**Target Code**	**Symbol**	**Target Output**
ISO 646	646	ISO 8859-1	8859	US ASCII
ISO 646de	646de	ISO 8859-1	8859	German
ISO 646da	646da	ISO 8859-1	8859	Danish
ISO 646en	646en	ISO 8859-1	8859	English ASCII
ISO 646es	646es	ISO 8859-1	8859	Spanish
ISO 646fr	646fr	ISO 8859-1	8859	French
ISO 646it	646it	ISO 8859-1	8859	Italian
ISO 646sv	646sv	ISO 8859-1	8859	Swedish

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

ISO 646 (US ASCII) to ISO 8859-1 For the conversion of ISO 646 to ISO 8859-1, all characters in ISO 646 can be mapped unchanged to ISO 8859-1

ISO 646de (GERMAN) to ISO 8859-1 For the conversion of ISO 646de to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646de	**ISO 8859-1**	**ISO 646de**	**ISO 8859-1**
100	247	173	344
133	304	174	366
134	326	175	374
135	334	176	337

ISO 646da (DANISH) to ISO 8859-1 For the conversion of ISO 646da to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646da	ISO 8859-1	ISO 646da	ISO 8859-1
133	306	173	346
134	330	174	370
135	305	175	345

ISO 646en (ENGLISH ASCII) to ISO 8859-1

For the conversion of ISO 646en to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed	
ISO 646en	ISO 8859-1
043	243

ISO 646es (SPANISH) to ISO 8859-1

For the conversion of ISO 646es to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646es	ISO 8859-1	ISO 646es	ISO 8859-1
100	247	173	260
133	241	174	361
134	321	175	347
135	277		

ISO 646fr (FRENCH) to ISO 8859-1

For the conversion of ISO 646fr to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646fr	ISO 8859-1	ISO 646fr	ISO 8859-1
043	243	173	351
100	340	174	371
133	260	175	350
134	347	176	250
135	247		

ISO 646it (ITALIAN) to ISO 8859-1

For the conversion of ISO 646it to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646it	ISO 8859-1	ISO 646it	ISO 8859-1
043	243	140	371
100	247	173	340
133	260	174	362
134	347	175	350
135	351	176	354

ISO 646sv (SWEDISH) to ISO 8859-1

For the conversion of ISO 646sv to ISO 8859-1, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 646sv	ISO 8859-1	ISO 646sv	ISO 8859-1
100	311	140	351
133	304	173	344
134	326	174	366
135	305	175	345
136	334	176	374

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_852 – code set conversion tables for MS 852 (MS-DOS Latin 2)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
MS 852	dos2	ISO 8859-2	iso2	ISO Latin 2
MS 852	dos2	MS 1250	win2	Windows Latin 2
MS 852	dos2	Mazovia	maz	Mazovia
MS 852	dos2	DHN	dhn	Dom Handlowy Nauki

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

MS 852 to ISO 8859-2 For the conversion of MS 852 to ISO 8859-2, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 852	ISO 8859-2	MS 852	ISO 8859-2
24-177	40	271-274	40
200	307	275	257
201	374	276	277
202	351	277-305	40
203	342	306	303
204	344	307	343
205	371	310-316	40
206	346	317	244
207	347	320	360
210	263	321	320
211	353	322	317
212	325	323	313
213	365	324	357
214	356	325	322

Conversions Performed			
MS 852	ISO 8859-2	MS 852	ISO 8859-2
215	254	326	315
216	304	327	316
217	306	330	354
220	311	331-334	40
221	305	335	336
222	345	336	331
223	364	337	40
224	366	340	323
225	245	341	337
226	265	342	324
227	246	343	321
230	266	344	361
231	326	345	362
232	334	346	251
233	253	347	271
234	273	350	300
235	243	351	332
236	327	352	340
237	350	353	333
240	341	354	375
241	355	355	335
242	363	356	376
243	372	357	264
244	241	360	255
245	261	361	275
246	256	362	262
247	276	363	267

Conversions Performed			
MS 852	ISO 8859-2	MS 852	ISO 8859-2
250	312	364	242
251	352	365	247
252	40	366	367
253	274	367	270
254	310	370	260
255	272	371	250
256-264	40	372	377
265	301	374	330
266	302	375	370
267	314	376	40
270	252		

MS 852 to MS 1250

For the conversion of MS 852 to MS 1250, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 852	MS 1250	MS 852	MS 1250
200	307	270	252
201	374	271-274	40
202	351	275	257
203	342	276	277
204	344	277-305	40
205	371	306	303
206	346	307	343
207	347	310-316	40
210	263	317	244
211	353	320	360
212	325	321	320
213	365	322	317

Conversions Performed			
MS 852	MS 1250	MS 852	MS 1250
214	356	323	313
215	217	324	357
216	304	325	322
217	306	326	315
220	311	327	316
221	305	330	354
222	345	331-334	40
223	364	335	336
224	366	336	331
225	274	337	40
226	276	340	323
227	214	341	337
230	234	342	324
231	326	343	321
232	334	344	361
233	215	345	362
234	235	346	212
235	243	347	232
236	327	350	300
237	350	351	332
240	341	352	340
241	355	353	333
242	363	354	375
243	372	355	335
244	245	356	376
245	271	357	264
246	216	360	255

Conversions Performed			
MS 852	MS 1250	MS 852	MS 1250
247	236	361	275
250	312	362	262
251	352	363	241
252	254	364	242
253	237	365	247
254	310	366	367
255	272	367	270
256	253	370	260
257	273	371	250
260-264	40	372	377
265	301	374	330
266	302	375	370
267	314	376	40

MS 852 to Mazovia

For the conversion of MS 852 to Mazovia, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 852	Mazovia	MS 852	Mazovia
205	40	246-247	40
206	215	250	220
210	222	251	221
212-213	40	253	246
215	240	254-270	40
217	225	275	241
220-226	40	276	247
227	230	306-336	40
230	236	340	243
233-234	40	342	40

Conversions Performed			
MS 852	Mazovia	MS 852	Mazovia
235	234	343	245
236-243	40	344	244
244	217	345-375	40
245	206		

MS 852 to DHN

For the conversion of MS 852 to DHN, all characters not in the following table are mapped unchanged.

Conversions Performed			
MS 852	DHN	MS 852	DHN
200-205	40	244	200
206	212	245	211
207	40	246-247	40
210	214	250	202
211-214	40	251	213
215	207	253	220
216	40	254-270	40
217	201	275	210
220-226	40	276	221
227	206	306-336	40
230	217	340	205
233-234	40	342	40
235	203	343	204
236-237	40	344	215
242	216	345-375	40
252	254		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目 iconv(1), iconv(3C), iconv(5)

名前 iconv_8859-1 – code set conversion tables for ISO 8859-1 (Latin 1)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
ISO 8859-1	8859	ISO 646	646	7 bit ASCII
ISO 8859-1	8859	ISO 646de	646de	German
ISO 8859-1	8859	ISO 646da	646da	Danish
ISO 8859-1	8859	ISO 646en	646en	English ASCII
ISO 8859-1	8859	ISO 646es	646es	Spanish
ISO 8859-1	8859	ISO 646fr	646fr	French
ISO 8859-1	8859	ISO 646it	646it	Italian
ISO 8859-1	8859	ISO 646sv	646sv	Swedish

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

ISO 8859-1 to ISO 646 (7-bit ASCII) For the conversion of ISO 8859-1 to ISO 646, all characters not in the following table are mapped unchanged.

```
Converted to Underscore '_' (137)
---------------------------------
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242 243 244 245 246 247
250 251 252 253 254 255 256 257
260 261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303 304 305 306 307
310 311 312 313 314 315 316 317
320 321 322 323 324 325 326 327
330 331 332 333 334 335 336 337
340 341 342 343 344 345 346 347
350 351 352 353 354 355 356 357
360 361 362 363 364 365 366 367
370 371 372 373 374 375 376 377
```

ISO 8859-1 to ISO 646de (GERMAN)

For the conversion of ISO 8859-1 to ISO 646de, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646de	ISO 8859-1	ISO 646de
247	100	337	176
304	133	344	173
326	134	366	174
334	135	374	175

```
Converted to Underscore '_' (137)
---------------------------------
100 133 134 135 173 174 175 176
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242 243 244 245 246
250 251 252 253 254 255 256 257
260 261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303     305 306 307
310 311 312 313 314 315 316 317
320 321 322 323 324 325     327
330 331 332 333     335 336 337
340 341 342 343     345 346 347
350 351 352 353 354 355 356 357
360 361 362 363 364 365     367
370 371 372 373     375 376 377
```

ISO 8859-1 to ISO 646da (DANISH)

For the conversion of ISO 8859-1 to ISO 646da, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646da	ISO 8859-1	ISO 646da
305	135	345	175
306	133	346	173
330	134	370	174

```
Converted to Underscore '_' (137)
---------------------------------
133 134 135 173 174 175
```

```
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242 243 244 245 246 247
250 251 252 253 254 255 256 257
260 261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303 304         307
310 311 312 313 314 315 316 317
320 321 322 323 324 325 326 327
    331 332 333 334 335 336 337
340 341 342 343 344         347
350 351 352 353 354 355 356 357
360 361 362 363 364 365 366 367
371 372 373 374         376 377
```

ISO 8859-1 to ISO 646en (ENGLISH ASCII)

For the conversion of ISO 8859-1 to ISO 646en, all characters not in the following tables are mapped unchanged.

Conversions Performed	
ISO 8859-1	**ISO 646en**
243	043

```
Converted to Underscore '_' (137)
---------------------------------
043
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242     244 245 246 247
250 251 252 253 254 255 256 257
260 261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303 304 305 306 307
310 311 312 313 314 315 316 317
320 321 322 323 324 325 326 327
330 331 332 333 334 335 336 337
340 341 342 343 344 345 346 347
350 351 352 353 354 355 356 357
360 361 362 363 364 365 366 367
370 371 372 373 374 375 376 377
```

ISO 8859-1 to ISO 646fr (FRENCH)

For the conversion of ISO 8859-1 to ISO 646fr, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646fr	ISO 8859-1	ISO 646fr
243	043	347	134
247	135	350	175
250	176	351	173
260	133	371	174
340	100		

```
Converted to Underscore '_' (137)
---------------------------------
043
100 133 134 135 173 174 175 176
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242     244 245 246
    251 252 253 254 255 256 257
    261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303 304 305 306 307
310 311 312 313 314 315 316 317
320 321 322 323 324 325 326 327
330 331 332 333 334 335 336 337
    341 342 343 344 345 346
        352 353 354 355 356 357
360 361 362 363 364 365 366 367
370     372 373 374 375 376 377
```

ISO 8859-1 to ISO 646it (ITALIAN)

For the conversion of ISO 8859-1 to ISO 646it, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646it	ISO 8859-1	ISO 646it
243	043	350	175
247	100	351	135
260	133	354	176

Conversions Performed			
ISO 8859-1	ISO 646it	ISO 8859-1	ISO 646it
340	173	362	174
347	134	371	140

```
Converted to Underscore '_' (137)
---------------------------------
043
100 133 134 135 173 174 175 176
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242     244 245 246
250 251 252 253 254 255 256 257
    261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303 304 305 306 307
310 311 312 313 314 315 316 317
320 321 322 323 324 325 326 327
330 331 332 333 334 335 336 337
    341 342 343 344 345 346
        352 353 354 355 356 357
360 361 362 363 364 365 366 367
370     372 373 374 375 376 377
```

ISO 8859-1 to ISO 646es (SPANISH)

For the conversion of ISO 8859-1 to ISO 646es, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646es	ISO 8859-1	ISO 646es
241	133	321	134
247	100	347	175
260	173	361	174
277	135		

```
Converted to Underscore '_' (137)
---------------------------------
100 133 134 135 173 174 175
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
```

```
240     242 243 244 245 246
250 251 252 253 254 255 256 257
    261 262 263 264 265 266 267
270 271 272 273 274 275 276
300 301 302 303 304 305 306 307
310 311 312 313 314 315 316 317
320     322 323 324 325 326 327
330 331 332 333 334 335 336 337
340 341 342 343 344 345 346
350 351 352 353 354 355 356 357
360     362 363 364 365 366 367
370 371 372 373 374 375 376 377
```

ISO 8859-1 to ISO 646sv (SWEDISH)

For the conversion of ISO 8859-1 to ISO 646sv, all characters not in the following tables are mapped unchanged.

Conversions Performed			
ISO 8859-1	ISO 646sv	ISO 8859-1	ISO 646sv
304	133	344	173
305	135	345	175
311	100	351	140
326	134	366	174
334	136	374	176

```
Converted to Underscore '_' (137)
---------------------------------
100 133 134 135 136 140
173 174 175 176
200 201 202 203 204 205 206 207
210 211 212 213 214 215 216 217
220 221 222 223 224 225 226 227
230 231 232 233 234 235 236 237
240 241 242 243 244 245 246 247
250 251 252 253 254 255 256 257
260 261 262 263 264 265 266 267
270 271 272 273 274 275 276 277
300 301 302 303         306 307
310     312 313 314 315 316 317
320 321 322 323 324 325     327
330 331 332 333     335 336 337
340 341 342 343         346 347
350     352 353 354 355 356 357
360 361 362 363 364 365     367
370 371 372 373     375 376 377
```

ファイル	`/usr/lib/iconv/*.so`	conversion modules
	`/usr/lib/iconv/*.t`	conversion tables
	`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables
関連項目	`iconv(1)`, `iconv(3C)`, `iconv(5)`	

名前 iconv_8859-2 – code set conversion tables for ISO 8859-2 (Latin 2)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	**Symbol**	**Target Code**	**Symbol**	**Target Output**
ISO 8859-2	iso2	MS 1250	win2	Windows Latin 2
ISO 8859-2	iso2	MS 852	dos2	MS-DOS Latin 2
ISO 8859-2	iso2	Mazovia	maz	Mazovia
ISO 8859-2	iso2	DHN	dhn	Dom Handlowy Nauki

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

ISO 8859-2 to MS 1250 For the conversion of ISO 8859-2 to MS 1250, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-2	**MS 1250**	**ISO 8859-2**	**MS 1250**
24	4	261	271
177-237	40	265	276
241	245	266	234
245	274	267	241
246	214	271	232
251	212	273	235
253	215	274	237
254	217	276	236
256	216	266	236

ISO 8859-2 to MS 852 For the conversion of ISO 8859-2 to MS 852, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-2	MS 852	ISO 8859-2	MS 852
24	4	316	327
177-237	40	317	322
240	377	320	321
241	244	321	343
242	364	322	325
243	235	323	340
244	317	324	342
245	225	325	212
246	227	326	231
247	365	327	236
250	371	330	374
251	346	331	336
252	270	332	351
253	233	333	353
254	215	334	232
255	360	335	355
256	246	336	335
257	275	337	341
260	370	340	352
261	245	341	240
262	362	342	203
263	210	343	307
264	357	344	204
265	226	345	222
266	230	346	206
267	363	347	207
270	367	350	237

Conversions Performed			
ISO 8859-2	MS 852	ISO 8859-2	MS 852
271	347	351	202
272	255	352	251
273	234	353	211
274	253	354	330
275	361	355	241
276	247	356	214
277	276	357	324
300	350	360	320
301	265	361	344
302	266	362	345
303	306	363	242
304	216	364	223
305	221	365	213
306	217	366	224
307	200	367	366
310	254	370	375
311	220	371	205
312	250	372	243
313	323	374	201
314	267	375	354
315	326	376	356
366	367		

ISO 8859-2 to Mazovia

For the conversion of ISO 8859-2 to Mazovia, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-2	Mazovia	ISO 8859-2	Mazovia
24	4	323	243

Conversions Performed			
ISO 8859-2	Mazovia	ISO 8859-2	Mazovia
177-237	40	324-325	40
240	377	326	231
241	217	327-333	40
242	40	334	232
243	234	335-336	40
244-245	40	337	341
246	230	340-341	40
247-253	40	342	203
254	240	343	40
255-256	40	344	204
257	241	345	40
260	370	346	215
261	206	347	207
262	40	350	40
263	222	351	202
264-265	40	352	221
266	236	353	211
267-273	40	354-355	40
274	246	356	214
275-276	40	357-360	40
277	247	361	244
300-303	40	362	40
304	216	363	242
305	40	364	223
306	225	365	40
307	200	366	224
310-311	40	367	366

Conversions Performed			
ISO 8859-2	Mazovia	ISO 8859-2	Mazovia
312	220	370-373	40
313-320	40	374	201
321	245	375-376	40
322	40		

ISO 8859-2 to DHN

For the conversion of ISO 8859-2 to DHN, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-2	DHN	ISO 8859-2	DHN
24	4	322	40
177-237	40	323	205
240	377	324-325	40
241	200	326	231
242	40	327-333	40
243	203	334	232
244-245	40	335-336	40
246	206	337	341
247-253	40	340	40
254	207	341	240
255-256	40	342-345	40
257	210	346	212
260	370	347-351	40
261	211	352	213
262	40	353-354	40
263	214	355	241
264-265	40	356-360	40
266	217	361	215
267-273	40	362	40

Conversions Performed			
ISO 8859-2	DHN	ISO 8859-2	DHN
274	220	363	216
275-276	40	364	223
277	221	365	40
300-305	40	366	224
306	201	367	366
307-311	40	370-371	40
312	202	372	243
313-320	40	373-376	40
321	204		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_8859-5 – code set conversion tables for ISO 8859-5 (Cyrillic)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
ISO 8859-5	iso5	KOI8-R	koi8	KOI8-R
ISO 8859-5	iso5	PC Cyrillic	alt	Alternative PC Cyrillic
ISO 8859-5	iso5	MS 1251	win5	Windows Cyrillic
ISO 8859-5	iso5	Mac Cyrillic	mac	Macintosh Cyrillic

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

ISO 8859-5 to KOI8-R For the conversion of ISO 8859-5 to KOI8-R, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-5	KOI8-R	ISO 8859-5	KOI8-R
24	4	320	301
241	263	321	302
242	261	322	327
243	262	323	307
244	264	324	304
245	265	325	305
246	266	327	332
247	267	330	311
250	270	331	312
251	271	332	313
252	272	333	314
253	273	334	315
254	274	335	316
256	276	336	317

Conversions Performed			
ISO 8859-5	KOI8-R	ISO 8859-5	KOI8-R
257	277	337	320
260	341	340	322
261	342	341	323
262	367	342	324
263	347	343	325
264	344	344	306
265	345	345	310
266	366	346	303
267	372	347	336
270	351	350	333
271	352	351	335
272	353	352	337
273	354	353	331
274	355	354	330
275	356	355	334
276	357	356	300
277	360	357	321
300	362	360	260
301	363	361	243
302	364	362	241
303	365	363	242
304	346	364	244
305	350	365	245
306	343	366	246
307	376	367	247
310	373	370	250
311	375	371	251

Conversions Performed			
ISO 8859-5	KOI8-R	ISO 8859-5	KOI8-R
312	377	372	252
313	371	373	253
314	370	374	254
315	374	375	255
316	340	376	256
317	361		

ISO 8859-5 to PC Cyrillic

For the conversion of ISO 8859-5 to PC Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-5	PC Cyrillic	ISO 8859-5	PC Cyrillic
24	4	307	227
200-240	40	310	230
241	360	311	231
242-254	40	312	232
255	260	313	233
256-257	40	314	234
260	200	315	235
261	201	316	236
262	202	317	237
263	203	320	240
264	204	321	241
265	205	322	242
266	206	323	243
267	207	324	244
270	210	325	245
271	211	326	246
272	212	327	247

Conversions Performed			
ISO 8859-5	PC Cyrillic	ISO 8859-5	PC Cyrillic
273	213	330	250
274	214	331	251
275	215	332	252
276	216	333	253
277	217	334	254
300	220	335	255
301	221	336	256
302	222	337	257
303	223	360-374	40
304	224	375	260
305	225	376	40
306	226	365	40

ISO 8859-5 to MS 1251

For the conversion of ISO 8859-5 to MS 1251, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-5	MS 1251	ISO 8859-5	MS 1251
24	4	317	337
200-237	40	320	340
241	250	321	341
242	200	322	342
243	201	323	343
244	252	324	344
245	275	325	345
246	262	326	346
247	257	327	347
250	243	330	350
251	212	331	351

Conversions Performed			
ISO 8859-5	MS 1251	ISO 8859-5	MS 1251
252	214	332	352
253	216	333	353
254	215	334	354
255	210	335	355
256	241	336	356
257	217	337	357
260	300	340	360
261	301	341	361
262	302	342	362
263	303	343	363
264	304	344	364
265	305	345	365
266	306	346	366
267	307	347	367
270	310	350	370
271	311	351	371
272	312	352	372
273	313	353	373
274	314	354	374
275	315	355	375
276	316	356	376
277	317	357	377
300	320	360	271
301	321	361	270
302	322	362	220
303	323	363	203
304	324	364	272

Conversions Performed			
ISO 8859-5	MS 1251	ISO 8859-5	MS 1251
305	325	365	276
306	326	366	263
307	327	367	277
310	330	370	274
311	331	371	232
312	332	372	234
313	333	373	236
314	334	374	235
315	335	375	210
316	336	376	242
376	331		

ISO 8859-5 to Mac Cyrillic

For the conversion of ISO 8859-5 to Mac Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
ISO 8859-5	Mac Cyrillic	ISO 8859-5	Mac Cyrillic
24	4	317	237
200-237	40	320	340
240	312	321	341
241	335	322	342
242	253	323	343
243	256	324	344
244	270	325	345
245	301	326	346
246	247	327	347
247	272	330	350
250	267	331	351
251	274	332	352

Conversions Performed			
ISO 8859-5	Mac Cyrillic	ISO 8859-5	Mac Cyrillic
252	276	333	353
253	40	334	354
254	315	335	355
255	40	336	356
256	330	337	357
257	332	340	360
260	200	341	361
261	201	342	362
262	202	343	363
263	203	344	364
264	204	345	365
265	205	346	366
266	206	347	367
267	207	350	370
270	210	351	371
271	211	352	372
272	212	353	373
273	213	354	374
274	214	355	375
275	215	356	376
276	216	357	337
277	217	360	334
300	220	361	336
301	221	362	254
302	222	363	257
303	223	364	271
304	224	365	317

Conversions Performed			
ISO 8859-5	Mac Cyrillic	ISO 8859-5	Mac Cyrillic
305	225	366	264
306	226	367	273
307	227	370	300
310	230	371	275
311	231	372	277
312	232	373	40
313	233	374	316
314	234	375	40
315	235	376	331
316	236		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_dhn – code set conversion tables for DHN (Dom Handlowy Nauki)

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
DHN	dhn	ISO 8859-2	iso2	ISO Latin 2
DHN	dhn	MS 1250	win2	Windows Latin 2
DHN	dhn	MS 852	dos2	MS-DOS Latin 2
DHN	dhn	Mazovia	maz	Mazovia

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

DHN to ISO 8859-2 For the conversion of DHN to ISO 8859-2, all characters not in the following table are mapped unchanged.

Conversions Performed			
DHN	ISO 8859-2	DHN	ISO 8859-2
24-177	40	222	40
200	241	223	364
201	306	224	366
202	312	225-230	40
203	243	231	326
204	321	232	334
205	323	233-237	40
206	246	240	341
207	254	241	355
210	257	242	363
211	261	243	372
212	346	244-340	40
213	352	341	337
214	263	342-365	40

Conversions Performed			
DHN	**ISO 8859-2**	**DHN**	**ISO 8859-2**
215	361	366	367
216	363	367	40
217	266	370	260
220	274	371-376	40
221	277		

DHN to MS 1250

For the conversion of DHN to MS 1250, all characters not in the following table are mapped unchanged.

Conversions Performed			
DHN	**MS 1250**	**DHN**	**MS 1250**
200	245	233-237	40
201	306	240	341
202	312	241	355
203	243	242	363
204	321	243	372
205	323	244-251	40
206	214	252	254
207	217	253-255	40
210	257	256	253
211	271	257	273
212	346	260-340	40
213	352	341	337
214	263	342-345	40
215	361	346	265
216	363	347-360	40
217	234	361	261
220	237	362-365	40
221	277	366	367

Conversions Performed			
DHN	MS 1250	DHN	MS 1250
222	40	367	40
223	364	370	260
224	366	371	40
225-230	40	372	267
231	326	373-376	40
232	334		

DHN to MS 852

For the conversion of DHN to MS 852, all characters not in the following table are mapped unchanged.

Conversions Performed			
DHN	MS 852	DHN	MS 852
200	244	212	206
201	217	213	251
202	250	214	210
203	235	215	344
204	343	216	242
205	340	217	230
206	227	220	253
207	215	221	276
210	275	222-375	40
211	245		

DHN to Mazovia

For the conversion of DHN to Mazovia, all characters not in the following table are mapped unchanged.

Conversions Performed			
DHN	Mazovia	DHN	Mazovia
200	217	212	215
201	225	213	221

Conversions Performed			
DHN	Mazovia	DHN	Mazovia
202	220	214	222
203	234	215	244
204	245	216	242
205	243	217	236
206	230	220	246
207	240	221	247
210	241	222-247	40
211	206		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_koi8-r – code set conversion tables for KOI8-R

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
KOI8-R	koi8	ISO 8859-5	iso5	ISO 8859-5 Cyrillic
KOI8-R	koi8	PC Cyrillic	alt	Alternative PC Cyrillic
KOI8-R	koi8	MS 1251	win5	Windows Cyrillic
KOI8-R	koi8	Mac Cyrillic	mac	Macintosh Cyrillic

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

KOI8-R to ISO 8859-5 For the conversion of KOI8-R to ISO 8859-5, all characters not in the following table are mapped unchanged.

Conversions Performed			
KOI8-R	ISO 8859-5	KOI8-R	ISO 8859-5
24	4	320	337
241	362	321	357
242	363	322	340
243	361	323	341
244	364	324	342
245	365	325	343
246	366	327	322
247	367	330	354
250	370	331	353
251	371	332	327
252	372	333	350
253	373	334	355
254	374	335	351
256	376	336	347

Conversions Performed			
KOI8-R	ISO 8859-5	KOI8-R	ISO 8859-5
257	377	337	352
260	360	340	316
261	242	341	260
262	243	342	261
263	241	343	306
264	244	344	264
265	245	345	265
266	246	346	304
267	247	347	263
270	250	350	305
271	251	351	270
272	252	352	271
273	253	353	272
274	254	354	273
275	255	355	274
276	256	356	275
277	257	357	276
300	356	360	277
301	320	361	317
302	321	362	300
303	346	363	301
304	324	364	302
305	325	365	303
306	344	366	266
307	323	367	262
310	345	370	314
311	330	371	313

Conversions Performed			
KOI8-R	ISO 8859-5	KOI8-R	ISO 8859-5
312	331	372	267
313	332	373	310
314	333	374	315
315	334	375	311
316	335	376	307
317	336		

KOI8-R to PC Cyrillic

For the conversion of KOI8-R to PC Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
KOI8-R	PC Cyrillic	KOI8-R	PC Cyrillic
24	4	333	350
200-242	40	334	355
243	361	335	351
244-254	40	336	347
255	260	337	352
256-262	40	340	236
263	360	341	200
264-274	40	342	201
275	260	343	226
276-277	40	344	204
300	356	345	205
301	240	346	224
302	241	347	203
303	346	350	225
304	244	351	210
305	245	352	211
306	344	353	212

Conversions Performed			
KOI8-R	PC Cyrillic	KOI8-R	PC Cyrillic
307	243	354	213
310	345	355	214
311	250	356	215
312	251	357	216
313	252	360	217
314	253	361	237
315	254	362	220
316	255	363	221
317	256	364	222
320	257	365	223
321	357	366	206
322	340	367	202
323	341	370	234
324	342	371	233
325	343	372	207
326	246	373	230
327	242	374	235
330	354	375	231
331	353	376	227
332	247		

KOI8-R to MS 1251

For the conversion of KOI8-R to MS 1251, all characters not in the following table are mapped unchanged.

Conversions Performed			
KOI8-R	MS 1251	KOI8-R	MS 1251
24	4	317	356
200-237	40	320	357
241	220	321	377

Conversions Performed			
KOI8-R	MS 1251	KOI8-R	MS 1251
242	203	322	360
243	270	323	361
244	272	324	362
245	276	325	363
246	263	326	346
247	277	327	342
250	274	330	374
251	232	331	373
252	234	332	347
253	236	333	370
254	235	334	375
255	210	335	371
256	242	336	367
257	237	337	372
260	271	340	336
261	200	341	300
262	201	342	301
263	250	343	326
264	252	344	304
265	275	345	305
266	262	346	324
267	257	347	303
270	243	350	325
271	212	351	310
272	214	352	311
273	216	353	312
274	215	354	313

Conversions Performed			
KOI8-R	MS 1251	KOI8-R	MS 1251
275	210	355	314
276	241	356	315
277	217	357	316
300	376	360	317
301	340	361	337
302	341	362	320
303	366	363	321
304	344	364	322
305	345	365	323
306	364	366	306
307	343	367	302
310	365	370	334
311	350	371	333
312	351	372	307
313	352	373	330
314	353	374	335
315	354	375	331
316	355	376	327
376	227		

KOI8-R to Mac Cyrillic

For the conversion of KOI8-R to Mac Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
KOI8-R	Mac Cyrillic	KOI8-R	Mac Cyrillic
24	4	317	356
200-237	40	320	357
240	312	321	337
241	254	322	360

Conversions Performed			
KOI8-R	Mac Cyrillic	KOI8-R	Mac Cyrillic
242	257	323	361
243	336	324	362
244	271	325	363
245	317	326	346
246	264	327	342
247	273	330	374
250	300	331	373
251	275	332	347
252	277	333	370
253	40	334	375
254	316	335	371
255	40	336	367
256	331	337	372
257	333	340	236
260	334	341	200
261	253	342	201
262	256	343	226
263	335	344	204
264	270	345	205
265	301	346	224
266	247	347	203
267	272	350	225
270	267	351	210
271	274	352	211
272	276	353	212
273	40	354	213
274	315	355	214

Conversions Performed			
KOI8-R	Mac Cyrillic	KOI8-R	Mac Cyrillic
275	40	356	215
276	330	357	216
277	332	360	217
300	376	361	237
301	340	362	220
302	341	363	221
303	366	364	222
304	344	365	223
305	345	366	206
306	364	367	202
307	343	370	234
310	365	371	233
311	350	372	207
312	351	373	230
313	352	374	235
314	353	375	231
315	354	376	227
316	355		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_mac_cyr – code set conversion tables for Macintosh Cyrillic

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
Mac Cyrillic	mac	ISO 8859-5	iso5	ISO 8859-5 Cyrillic
Mac Cyrillic	mac	KOI8-R	koi8	KOI8-R
Mac Cyrillic	mac	PC Cyrillic	alt	Alternative PC Cyrillic
Mac Cyrillic	mac	MS 1251	win5	Windows Cyrillic

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

Mac Cyrillic to ISO 8859-5 For the conversion of Mac Cyrillic to ISO 8859-5, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mac Cyrillic	ISO 8859-5	Mac Cyrillic	ISO 8859-5
24	4	276	252
200	260	277	372
201	261	300	370
202	262	301	245
203	263	302-311	40
204	264	312	240
205	265	313	242
206	266	314	362
207	267	315	254
210	270	316	374
211	271	317	365
212	272	320-327	40
213	273	330	256
214	274	331	376

Conversions Performed			
Mac Cyrillic	ISO 8859-5	Mac Cyrillic	ISO 8859-5
215	275	332	257
216	276	333	377
217	277	334	360
220	300	335	241
221	301	336	361
222	302	337	357
223	303	340	320
224	304	341	321
225	305	342	322
226	306	343	323
227	307	344	324
230	310	345	325
231	311	346	326
232	312	347	327
233	313	350	330
234	314	351	331
235	315	352	332
236	316	353	333
237	317	354	334
240-246	40	355	335
247	246	356	336
250-252	40	357	337
253	242	360	340
254	362	361	341
255	40	362	342
256	243	363	343
257	363	364	344

Conversions Performed			
Mac Cyrillic	ISO 8859-5	Mac Cyrillic	ISO 8859-5
260-263	40	365	345
264	366	366	346
265-266	40	367	347
267	250	370	350
270	244	371	351
271	364	372	352
272	247	373	353
273	367	374	354
274	251	375	355
275	371	376	356
375	370		

Mac Cyrillic to KOI8-R

For the conversion of Mac Cyrillic to KOI8-R, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mac Cyrillic	KOI8-R	Mac Cyrillic	KOI8-R
24	4	276	272
200	341	277	252
201	342	300	250
202	367	301	265
203	347	302-311	40
204	344	312	240
205	345	313	261
206	366	314	241
207	372	315	274
210	351	316	254
211	352	317	245
212	353	320-327	40

Conversions Performed			
Mac Cyrillic	**KOI8-R**	**Mac Cyrillic**	**KOI8-R**
213	354	330	276
214	355	331	256
215	356	332	277
216	357	333	257
217	360	334	260
220	362	335	263
221	363	336	243
222	364	337	321
223	365	340	301
224	346	341	302
225	350	342	327
226	343	343	307
227	376	344	304
230	373	345	305
231	375	346	326
232	377	347	332
233	371	350	311
234	370	351	312
235	374	352	313
236	340	353	314
237	361	354	315
240-246	40	355	316
247	266	356	317
250-252	40	357	320
253	261	360	322
254	241	361	323
255	40	362	324

Conversions Performed			
Mac Cyrillic	KOI8-R	Mac Cyrillic	KOI8-R
256	262	363	325
257	242	364	306
260-263	40	365	310
264	246	366	303
265-266	40	367	336
267	270	370	333
270	264	371	335
271	244	372	337
272	267	373	331
273	247	374	330
274	271	375	334
275	251	376	300
375	370		

Mac Cyrillic to PC Cyrillic

For the conversion of Mac Cyrillic to PC Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mac Cyrillic	PC Cyrillic	Mac Cyrillic	PC Cyrillic
24	4	355	255
240-334	40	356	256
335	360	357	257
336	361	360	340
337	357	361	341
340	240	362	342
341	241	363	343
342	242	364	344
343	243	365	345
344	244	366	346

Conversions Performed			
Mac Cyrillic	**PC Cyrillic**	**Mac Cyrillic**	**PC Cyrillic**
345	245	367	347
346	246	370	350
347	247	371	351
350	250	372	352
351	251	373	353
352	252	374	354
353	253	375	355
354	254	376	356
303	366		

Mac Cyrillic to MS 1251

For the conversion of Mac Cyrillic to MS 1251, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mac Cyrillic	**MS 1251**	**Mac Cyrillic**	**MS 1251**
24	4	255	40
200	300	256	201
201	301	257	203
202	302	260-263	40
203	303	264	263
204	304	266	264
205	305	267	243
206	306	270	252
207	307	271	272
210	310	272	257
211	311	273	277
212	312	274	212
213	313	275	232
214	314	276	214

Conversions Performed			
Mac Cyrillic	MS 1251	Mac Cyrillic	MS 1251
215	315	277	234
216	316	300	274
217	317	301	275
220	320	302	254
221	321	303-306	40
222	322	307	253
223	323	310	273
224	324	311	205
225	325	312	240
226	326	313	200
227	327	314	220
230	330	315	215
231	331	316	235
232	332	317	276
233	333	320	226
234	334	321	227
235	335	322	223
236	336	323	224
237	337	324	221
240	206	325	222
241	260	326	40
242	245	327	204
243	40	330	241
244	247	331	242
245	267	332	217
246	266	333	237
247	262	334	271

Conversions Performed			
Mac Cyrillic	**MS 1251**	**Mac Cyrillic**	**MS 1251**
250	256	335	250
252	231	336	270
253	200	337	377
254	220	362	324

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_maz – code set conversion tables for Mazovia

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
Mazovia	maz	ISO 8859-2	iso2	ISO Latin 2
Mazovia	maz	MS 1250	win2	Windows Latin 2
Mazovia	maz	MS 852	dos2	MS-DOS Latin 2
Mazovia	maz	DHN	dhn	Dom Hanlowy Nauki

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

Mazovia to ISO 8859-2 For the conversion of Mazovia to ISO 8859-2, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mazovia	ISO 8859-2	Mazovia	ISO 8859-2
24–177	40	230	246
200	307	231	326
201	374	232	334
202	351	233	40
203	342	234	243
204	344	235	40
205	40	236	266
206	261	237	40
207	347	240	254
210	40	241	257
211	353	242	363
212-213	40	243	323
214	356	244	361
215	346	245	321

Conversions Performed			
Mazovia	**ISO 8859-2**	**Mazovia**	**ISO 8859-2**
216	304	246	274
217	241	247	277
220	312	250-340	40
221	352	341	337
222	263	342-365	40
223	364	366	367
224	366	367	40
225	306	370	260
226-227	40	371-376	40
256	201		

Mazovia to MS 1250

For the conversion of Mazovia to MS 1250, all characters not in the following table are mapped unchanged.

Mazovia	**MS 1250**	**Mazovia**	**MS 1250**
200	307	236	234
201	374	237	40
202	351	240	217
203	342	241	257
204	344	242	363
205	40	243	323
206	271	244	361
207	347	245	321
210	40	246	237
211	353	247	277
212-213	40	250-251	40
214	356	252	254
215	346	253-255	40
216	304	256	253

Mazovia	MS 1250	Mazovia	MS 1250
217	245	257	273
220	312	260-340	40
221	352	341	337
222	263	342-345	40
223	364	346	265
224	366	347-360	40
225	306	361	261
226-227	40	362-365	0
230	214	366	367
231	326	367	40
232	334	370	260
233	40	371	40
234	243	372	267
235	40	373-376	40
274	212		

Mazovia to MS 852

For the conversion of Mazovia to MS 852, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mazovia	MS 852	Mazovia	MS 852
205	40	234	235
206	245	235	40
210-213	40	236	230
215	206	237	40
217	244	240	215
220	250	241	275
221	251	243	340
222	210	244	344
225	217	245	343

Conversions Performed			
Mazovia	**MS 852**	**Mazovia**	**MS 852**
226-227	40	246	253
230	227	247	276
233	40	250-375	40
227	327		

Mazovia to DHN

For the conversion of Mazovia to DHN, all characters not in the following table are mapped unchanged.

Conversions Performed			
Mazovia	**DHN**	**Mazovia**	**DHN**
200-205	40	234	203
206	211	236	217
207-214	40	240	207
215	212	241	210
216	40	242	216
217	200	243	205
220	202	244	215
221	214	246	220
225	201	247	221
230	206		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_pc_cyr – code set conversion tables for Alternative PC Cyrillic

機能説明 The following code set conversions are supported:

Code Set Conversions Supported				
Code	Symbol	Target Code	Symbol	Target Output
PC Cyrillic	alt	ISO 8859-5	iso5	ISO 8859-5 Cyrillic
PC Cyrillic	alt	KOI8-R	koi8	KOI8-R
PC Cyrillic	alt	MS 1251	win5	Windows Cyrillic
PC Cyrillic	alt	Mac Cyrillic	mac	Macintosh Cyrillic

Conversions The conversions are performed according to the following tables. All values in the tables are given in octal.

PC Cyrillic to ISO 8859-5 For the conversion of PC Cyrillic to ISO 8859-5, all characters not in the following table are mapped unchanged.

Conversions Performed			
PC Cyrillic	ISO 8859-5	PC Cyrillic	ISO 8859-5
24	4	231	311
200	260	232	312
201	261	233	313
202	262	234	314
203	263	235	315
204	264	236	316
205	265	237	317
206	266	240	320
207	267	241	321
210	270	242	322
211	271	243	323
212	272	244	324
213	273	245	325
214	274	246	326

Conversions Performed			
PC Cyrillic	ISO 8859-5	PC Cyrillic	ISO 8859-5
215	275	247	327
216	276	250	330
217	277	251	331
220	300	252	332
221	301	253	333
222	302	254	334
223	303	255	335
224	304	256	336
225	305	257	337
226	306	260-337	255
227	307	360	241
230	310	362-376	255

PC Cyrillic to KOI8-R

For the conversion of PC Cyrillic to KOI8-R, all characters not in the following table are mapped unchanged.

Conversions Performed			
PC Cyrillic	KOI8-R	PC Cyrillic	KOI8-R
24	4	242	327
200	341	243	307
201	342	244	304
202	367	245	305
203	347	246	326
204	344	247	332
205	345	250	311
206	366	251	312
207	372	252	313
210	351	253	314
211	352	254	315

Conversions Performed			
PC Cyrillic	KOI8-R	PC Cyrillic	KOI8-R
212	353	255	316
213	354	256	317
214	355	257	320
215	356	260-337	255
216	357	340	322
217	360	341	323
220	362	342	324
221	363	343	325
222	364	344	306
223	365	345	310
224	346	346	303
225	350	347	336
226	343	350	333
227	376	351	335
230	373	352	337
231	375	353	331
232	377	354	330
233	371	355	334
234	370	356	300
235	374	357	321
236	340	360	263
237	361	361	243
240	301	362-376	255
241	302		

PC Cyrillic to MS 1251

For the conversion of PC Cyrillic to MS 1251, all characters not in the following table are mapped unchanged.

Conversions Performed			
PC Cyrillic	**MS 1251**	**PC Cyrillic**	**MS 1251**
24	4	242	342
200	300	243	343
201	301	244	344
202	302	245	345
203	303	246	346
204	304	247	347
205	305	250	350
206	306	251	351
207	307	252	352
210	310	253	353
211	311	254	354
212	312	255	355
213	313	256	356
214	314	257	357
215	315	260-337	210
216	316	340	360
217	317	341	361
220	320	342	362
221	321	343	363
222	322	344	364
223	323	345	365
224	324	346	366
225	325	347	367
226	326	350	370
227	327	351	371
230	330	352	372
231	331	353	373

Conversions Performed			
PC Cyrillic	MS 1251	PC Cyrillic	MS 1251
232	332	354	374
233	333	355	375
234	334	356	376
235	335	357	377
236	336	360	250
237	337	361	270
240	340	362-376	210
241	341		

PC Cyrillic to Mac Cyrillic

For the conversion of PC Cyrillic to Mac Cyrillic, all characters not in the following table are mapped unchanged.

Conversions Performed			
PC Cyrillic	Mac Cyrillic	PC Cyrillic	Mac Cyrillic
24	4	341	361
240	340	342	362
241	341	343	363
242	342	344	364
243	343	345	365
244	344	346	366
245	345	347	367
246	346	350	370
247	347	351	371
250	350	352	372
251	351	353	373
252	352	354	374
253	353	355	375
254	354	356	376
255	355	357	337

Conversions Performed			
PC Cyrillic	**Mac Cyrillic**	**PC Cyrillic**	**Mac Cyrillic**
256	356	360	335
257	357	361	336
260-337	40	362-376	40
340	360		

ファイル

`/usr/lib/iconv/*.so`	conversion modules
`/usr/lib/iconv/*.t`	conversion tables
`/usr/lib/iconv/iconv_data`	list of conversions supported by conversion tables

関連項目

`iconv(1)`, `iconv(3C)`, `iconv(5)`

名前 iconv_unicode – code set conversion tables for Unicode

機能説明 The following code set conversions are supported:

CODE SET CONVERSIONS SUPPORTED

FROM Code Set Code	FROM Filename Element	TO Code Set Target Code	TO Filename Element
ISO 8859-1 (Latin 1)	8859-1	UTF-8	UTF-8
ISO 8859-2 (Latin 2)	8859-2	UTF-8	UTF-8
ISO 8859-3 (Latin 3)	8859-3	UTF-8	UTF-8
ISO 8859-4 (Latin 4)	8859-4	UTF-8	UTF-8
ISO 8859-5 (Cyrillic)	8859-5	UTF-8	UTF-8
ISO 8859-6 (Arabic)	8859-6	UTF-8	UTF-8
ISO 8859-7 (Greek)	8859-7	UTF-8	UTF-8
ISO 8859-8 (Hebrew)	8859-8	UTF-8	UTF-8
ISO 8859-9 (Latin 5)	8859-9	UTF-8	UTF-8
ISO 8859-10 (Latin 6)	8859-10	UTF-8	UTF-8
Japanese EUC	eucJP	UTF-8	UTF-8
Chinese/PRC EUC (GB 2312-1980)	gb2312	UTF-8	UTF-8
ISO-2022	iso2022	UTF-8	UTF-8
Korean EUC	ko_KR-euc	Korean UTF-8	ko_KR-UTF-8
ISO-2022-KR	ko_KR-iso2022-7	Korean UTF-8	ko_KR_UTF-8
Korean Johap (KS C 5601-1987)	ko_KR-johap	Korean UTF-8	ko_KR-UTF-8
Korean Johap (KS C 5601-1992)	ko_KR-johap92	Korean UTF-8	ko_KR-UTF-8
Korean UTF-8	ko_KR-UTF-8	Korean EUC	ko_KR-euc
Korean UTF-8	ko_KR-UTF-8	Korean Johap (KS C 5601-1987)	ko_KR-johap
Korean UTF-8	ko_KR-UTF-8	Korean Johap (KS C 5601-1992)	ko_KR-johap92
KOI8-R (Cyrillic)	KOI8-R	UCS-2	UCS-2
KOI8-R (Cyrillic)	KOI8-R	UTF-8	UTF-8
PC Kanji (SJIS)	PCK	UTF-8	UTF-8
PC Kanji (SJIS)	SJIS	UTF-8	UTF-8
UCS-2	UCS-2	KOI8-R (Cyrillic)	KOI8-R
UCS-2	UCS-2	UCS-4	UCS-4

CODE SET CONVERSIONS SUPPORTED

FROM Code Set Code	FROM Filename	TO Code Set Target Code	TO Filename

	Element		Element
UCS-2	UCS-2	UTF-7	UTF-7
UCS-2	UCS-2	UTF-8	UTF-8
UCS-4	UCS-4	UCS-2	UCS-2
UCS-4	UCS-4	UTF-16	UTF-16
UCS-4	UCS-4	UTF-7	UTF-7
UCS-4	UCS-4	UTF-8	UTF-8
UTF-16	UTF-16	UCS-4	UCS-4
UTF-16	UTF-16	UTF-8	UTF-8
UTF-7	UTF-7	UCS-2	UCS-2
UTF-7	UTF-7	UCS-4	UCS-4
UTF-7	UTF-7	UTF-8	UTF-8
UTF-8	UTF-8	ISO 8859-1 (Latin 1)	8859-1
UTF-8	UTF-8	ISO 8859-2 (Latin 2)	8859-2
UTF-8	UTF-8	ISO 8859-3 (Latin 3)	8859-3
UTF-8	UTF-8	ISO 8859-4 (Latin 4)	8859-4
UTF-8	UTF-8	ISO 8859-5 (Cyrillic)	8859-5
UTF-8	UTF-8	ISO 8859-6 (Arabic)	8859-6
UTF-8	UTF-8	ISO 8859-7 (Greek)	8859-7
UTF-8	UTF-8	ISO 8859-8 (Hebrew)	8859-8
UTF-8	UTF-8	ISO 8859-9 (Latin 5)	8859-9
UTF-8	UTF-8	ISO 8859-10 (Latin 6)	8859-10
UTF-8	UTF-8	Japanese EUC	eucJP
UTF-8	UTF-8	Chinese/PRC EUC (GB 2312-1980)	gb2312
UTF-8	UTF-8	ISO-2022	iso2022
UTF-8	UTF-8	KOI8-R (Cyrillic)	KOI8-R
UTF-8	UTF-8	PC Kanji (SJIS)	PCK
UTF-8	UTF-8	PC Kanji (SJIS)	SJIS
UTF-8	UTF-8	UCS-2	UCS-2
UTF-8	UTF-8	UCS-4	UCS-4
UTF-8	UTF-8	UTF-16	UTF-16
UTF-8	UTF-8	UTF-7	UTF-7
UTF-8	UTF-8	Chinese/PRC EUC (GB 2312-1980)	zh_CN.euc

CODE SET CONVERSIONS SUPPORTED

FROM Code Set Code	FROM Filename Element	TO Code Set Target Code	TO Filename Element
UTF-8	UTF-8	ISO 2022-CN	zh_CN.iso2022-7
UTF-8	UTF-8	Chinese/Taiwan Big5	zh_TW-big5
UTF-8	UTF-8	Chinese/Taiwan EUC (CNS 11643-1992)	zh_TW-euc

UTF-8	UTF-8	ISO 2022-TW	zh_TW-iso2022-7
Chinese/PRC EUC (GB 2312-1980)	zh_CN.euc	UTF-8	UTF-8
ISO 2022-CN	zh_CN.iso2022-7	UTF-8	UTF-8
Chinese/Taiwan Big5	zh_TW-big5	UTF-8	UTF-8
Chinese/Taiwan EUC (CNS 11643-1992)	zh_TW-euc	UTF-8	UTF-8
ISO 2022-TW	zh_TW-iso2022-7	UTF-8	UTF-8

使用例

例 1 The library module filename

In the conversion library, `/usr/lib/iconv` (see iconv(3C)), the library module filename is composed of two symbolic elements separated by the percent sign (`%`). The first symbol specifies the code set that is being converted; the second symbol specifies the *target code*, that is, the code set to which the first one is being converted.

In the conversion table above, the first symbol is termed the “FROM Filename Element”. The second symbol, representing the target code set, is the “TO Filename Element”.

For example, the library module filename to convert from the *Korean EUC* code set to the *Korean UTF-8* code set is

```
ko_KR-euc%ko_KR-UTF-8
```

ファイル

`/usr/lib/iconv/*.so` conversion modules

関連項目

iconv(1), iconv(3C), iconv(5)

Chernov, A., Registration of a Cyrillic Character Set, RFC 1489, RELCOM Development Team, July 1993.

Chon, K., H. Je Park, and U. Choi, Korean Character Encoding for Internet Messages, RFC 1557, Solvit Chosun Media, December 1993.

Goldsmith, D., and M. Davis, UTF-7 – A Mail-Safe Transformation Format of Unicode, RFC 1642, Taligent, Inc., July 1994.

Lee, F., HZ – A Data Format for Exchanging Files of Arbitrarily Mixed Chinese and ASCII characters, RFC 1843, Stanford University, August 1995.

Murai, J., M. Crispin, and E. van der Poel, Japanese Character Encoding for Internet Messages, RFC 1468, Keio University, Panda Programming, June 1993.

Nussbacher, H., and Y. Bourvine, Hebrew Character Encoding for Internet Messages, RFC 1555, Israeli Inter-University, Hebrew University, December 1993.

Ohta, M., Character Sets ISO-10646 and ISO-10646-J-1, RFC 1815, Tokyo Institute of Technology, July 1995.

Ohta, M., and K. Handa, ISO-2022-JP-2: Multilingual Extension of ISO-2022-JP, RFC 1554, Tokyo Institute of Technology, December 1993.

Reynolds, J., and J. Postel, ASSIGNED NUMBERS, RFC 1700, University of Southern California/Information Sciences Institute, October 1994.

Simonson, K., Character Mnemonics & Character Sets, RFC 1345, Rationel Almen Planlaegning, June 1992.

Spinellis, D., Greek Character Encoding for Electronic Mail Messages, RFC 1947, SENA S.A., May 1996.

The Unicode Consortium, The Unicode Standard, Version 2.0, Addison Wesley Developers Press, July 1996.

Wei, Y., Y. Zhang, J. Li, J. Ding, and Y. Jiang, ASCII Printable Characters-Based Chinese Character Encoding for Internet Messages, RFC 1842, AsiaInfo Services Inc., Harvard University, Rice University, University of Maryland, August 1995.

Yergeau, F., UTF-8, a transformation format of Unicode and ISO 10646, RFC 2044, Alis Technologies, October 1996.

Zhu, H., D. Hu, Z. Wang, T. Kao, W. Chang, and M. Crispin, Chinese Character Encoding for Internet Messages, RFC 1922, Tsinghua University, China Information Technology Standardization Technical Committee (CITS), Institute for Information Industry (III), University of Washington, March 1996.

注意事項

ISO 8859 character sets using Latin alphabetic characters are distinguished as follows:

`ISO 8859-1 (Latin 1)` For most West European languages, including:

Albanian	Finnish	Italian
Catalan	French	Norwegian
Danish	German	Portuguese
Dutch	Galician	Spanish
English	Irish	Swedish
Faeroese	Icelandic	

`ISO 8859-2 (Latin 2)` For most Latin-written Slavic and Central European languages:

Czech	Polish	Slovak

	German	Rumanian	Slovene
	Hungarian	Croatian	

ISO 8859-3 (Latin 3)	Popularly used for Esperanto, Galician, Maltese, and Turkish.
ISO 8859-4 (Latin 4)	Introduces letters for Estonian, Latvian, and Lithuanian. It is an incomplete predecessor of ISO 8859-10 (Latin 6).
ISO 8859-9 (Latin 5)	Replaces the rarely needed Icelandic letters in ISO 8859-1 (Latin 1) with the Turkish ones.
ISO 8859-10 (Latin 6)	Adds the last Inuit (Greenlandic) and Sami (Lappish) letters that were not included in ISO 8859-4 (Latin 4) to complete coverage of the Nordic area.

名前

ieee802.11 – 802.11 kernel statistics

機能説明

This page describes the kernel statistics that can be used to monitor attributes specific to the 802.11 physical layer. These statistics can be retrieved using kstat(1M). Not all 802.11 devices will support all statistics.

tx_frags	Count of data and management fragments transmitted.
rx_frags	Count of data and management fragments received.
rx_dups	Count of duplicate frames received. Duplicates are determined by the sequence control field.
mcast_tx	Count of broadcast and multicast frames transmitted.
mcast_rx	Count of broadcast and multicast frames received.
tx_failed	Count of frames that could not be transmitted due to the retransmission limit being reached.
tx_retrans	Count of frames successfully retransmitted after one or more retransmissions.
tx_reretrans	Count of frames successfully retransmitted after more than one retransmission.
rts_success	Count of times a CTS was received in response to an RTS.
rts_failure	Count of times a CTS was not received in response to an RTS.
ack_failure	Count of times an ACK was expected but was not received.
fcs_errors	Count of frames received with FCS errors.
wep_errors	Count of frames received with the WEP bit set but that either should not have been encrypted or that were discarded due to WEP not being supported.

関連項目

kstat(1M)

名前

ieee802.3, cap_autoneg, cap_1000fdx, cap_1000hdx, cap_100fdx, cap_100hdx, cap_10fdx, cap_10hdx, cap_rem_fault, cap_pause, cap_asym_pause, adv_cap_autoneg, adv_cap_1000fdx, adv_cap_1000hdx, adv_cap_100fdx, adv_cap_100hdx, adv_cap_10fdx, adv_cap_10hdx, adv_cap_pause, adv_cap_asym_pause, adv_rem_fault, lp_cap_autoneg, lp_cap_1000fdx, lp_cap_1000hdx, lp_cap_100fdx, lp_cap_100hdx, lp_cap_10fdx, lp_cap_10hdx, lp_cap_pause, lp_cap_asym_pause, lp_rem_fault, xcvr_addr, xcvr_id, xcvr_inuse, link_up, link_duplex, link_tx_pause, link_rx_pause – Ethernet mii kstat and dladm parameters

機能説明

This page describes the kernel statistics and the `dladm(1M)` configuration parameters used to monitor and configure the Ethernet physical layer.

The `cap_*` parameters exist in the kernel statistics for an Ethernet device. The parameters describe the maximum capability of a device. When the value of a statistic is `1`, the device has the capability described. When the value is `0`, the device does not have the capability.

The exceptions to this rule are the `cap_asym_pause` and `cap_pause` parameters which are explained later in this page.

`cap_autoneg`	Capable of auto-negotiation
`cap_1000fdx`	Capable of 1000 full duplex operation
`cap_1000hdx`	Capable of 1000 half duplex operation
`cap_100fdx`	Capable of 100 full duplex operation
`cap_100hdx`	Capable of 100 half duplex operation
`cap_10fdx`	Capable of 10 full duplex operation
`cap_10hdx`	Capable of 10 half duplex operation
`cap_rem_fault`	Capable of reporting locally detected faults to link partner

The `adv_cap_*` parameters exist in the kernel statistics and represent a mirror image of the `dladm adv_*_cap` parameter list for an Ethernet device. The `dladm adv_*_cap` tuning parameters allow fine grain control of the Ethernet device physical layer. The parameters are also a subset of the `cap_*` statistics. If the `cap_*` value is `0`, the corresponding `adv_cap_*` must also be `0`. The exceptions to this rule are the `adv_cap_asym_pause` and `adv_cap_pause` parameters.

When auto-negotiation is enabled, the `adv_*_cap` statistics show which capabilities are advertised to the link partner. When auto-negotiation is disabled in *forced mode*, the statistics precisely show how a link should function and that it must be matched on the link partner to achieve a valid link up.

Statistics with values other than `0` and `1` are also described in the following.

`adv_cap_autoneg` Advertise auto-negotiation capability

`adv_cap_1000fdx` Advertise 1000 full duplex capability

`adv_cap_1000hdx` Advertise 1000 half duplex capability

`adv_cap_100fdx` Advertise 100 full duplex capability

`adv_cap_100hdx` Advertise 100 half duplex capability

`adv_cap_10fdx` Advertise 10 full duplex capability

`adv_cap_10hdx` Advertise 10 half duplex capability

`adv_rem_fault` Fault value reported by the local system to the peer

- `0` Link is good
- `1` Fault

The `lp_cap_*` parameters exist as kernel statistics for an Ethernet device. The statistics are the advertised capabilities provided by the link partner on completion of auto-negotiation. If the capabilities match the capabilities provided in the local advertisement, the link can proceed to a link up state. If no match is found, the link remains down. In two other instances, `lp_cap_*` values might all be zero: (1) when a cable is not present and (2) when forced mode is enabled.

`lp_cap_autoneg` Link partner advertises auto-negotiation capability

`lp_cap_1000fdx` Link partner advertises 1000 full duplex capability

`lp_cap_1000hdx` Link partner advertises 1000 half duplex capability

`lp_cap_100fdx` Link partner advertises 100 full duplex capability

`lp_cap_100hdx` Link partner advertises 100 half duplex capability

`lp_cap_10fdx` Link partner advertises 10 full duplex capability

`lp_cap_10hdx` Link partner advertises 10 half duplex capability

`lp_rem_fault` Fault value the remote system reports

- `0` Link is good
- `1` Fault

The `xcvr_*` kernel statistics provide information about the physical layer device that is in use.

`xcvr_addr` MII address in the `0` to `31` range of the physical layer device in use for a given Ethernet device

`xcvr_id` MII transceiver manufacturer and device ID

`xcvr_inuse` MII transceiver type, based on the following list:

0 other	Undefined
1 none	MII present, but nothing connected
2 10Mb/s	10Mb/s Manchester encoding
3 100BaseT4	100 Mb/s 8B/6T
4 100BaseX	100 Mb/s 4B/5B
5 100BaseT2	100 Mb/s PAM5X5
6 1000BaseX	1000 Mb/s 8B/10B
7 1000BaseT	1000 Mb/s 4D-PAM5

The above values define maximum capability. In many cases, lower speeds can occur. The `cap_*` statistics must be viewed to establish the range of capability.

The `link_*` kernel statistics show the link state at the local end of the connection.

`link_up`	1	Link is up
	0	Link is down
`link_duplex`	2	Full duplex link
	1	Half duplex link
	0	Unknown

The `cap_asym_pause`, `cap_pause`, `adv_cap_asym_pause`, and `adv_cap_pause` parameters do not follow the rules of other `cap_*` and `adv_cap_*` kstats or parameters. The `cap_*pause` kstats provide information about the capabilities supported by the device and constrain the values that may be set to the corresponding `adv_cap_*pause` parameters.

`cap_pause`	Symmetric pause capability.
`cap_asym_pause`	Asymmetric pause capability.

The `adv_cap_pause` and `adv_cap_asym_pause` statistics are limited by the available settings for `cap_pause` and `cap_asym_pause`. These statistics are read-only values whose settings may be administratively controlled by setting the flowctrl property supported by `dladm(1M)`. For a device that is fully capable of pausing both Rx (receive) and Tx (transmit) operations, the settings available are defined in the truth table that follows the `adv_cap_pause` and `adv_cap_asm_pause` parameter descriptions below.

`adv_cap_pause`	When adv_cap_pause is 1, the device can both assert and respond to flow control. This is the pre-Gigabit, symmetric mode of

operation, and implies a full (both send and receive) implementation of the PAUSE mechanism within the device. In addition, if adv_cap_asym_pause is 1, the device can operate either symmetrically or asymmetrically in either direction.

If adv_cap_pause is 0, advertised, flow-control behavior is determined by adv_cap_asym_pause. If the value of adv_cap_asym_pause is 1, the device can assert flow control, but cannot resend.

No flow control is available when bot adv_cap_pause and adv_cap_asym_pause are 0.

`adv_cap_asym_pause` Asymmetric pause capability.

The `cap_asym_pause` and `cap_pause` statistics show the capability of a device and also limit the legal setting for `adv_cap_asym_pause` and `adv_cap_pause`. The following truth table describes the available `adv_cap_asym_pause` and `adv_cap_pause` settings limited by `cap_asym_pause` and `cap_pause` `statistics`. The abbreviations below are used in the table.

CA `cap_asym_pause`

CP `cap_pause`

AA `adv_cap_asym_pause`

AP `adv_cap_pause`

CP	CA	AP	AA	Description
0	0	0	0	No pause in use.
0	0	x	x	Device not pause capable, cannot set.
0	1	0	0	Asymmetric Rx pause capable, but not advertised.
0	1	0	1	Asymmetric Rx pause capable and advertised.
0	1	1	0	Asymmetric Rx pause capable, but not advertised. Not capable of symmetric pause.
0	1	1	1	Asymmetric Rx pause capable and advertised. No symmetric pause capability or asymmetric Tx pause.
1	0	0	0	Symmetric pause capable, but not advertised.
1	0	0	1	Symmetric pause capable, advertising asymmetric Rx pause only.
1	0	1	0	Symmetric pause capable, advertising symmetric Rx and Tx pause capability.

1	0	1	1	Symmetric pause capable and advertised.
1	1	0	0	Symmetric and asymmetric pause capable, but not advertised.
1	1	0	1	Symmetric and asymmetric Tx pause capable. Only asymmetric Tx pause advertised.
1	1	1	0	Symmetric and symmetric Tx pause capable. Only symmetric pause advertised.
1	1	1	1	Asymmetric Tx pause capable and advertised.

In the cases above, an error is posted when a device driver cannot advertise. A new setting is ignored and values revert to the previous setting.

The `lp_cap_pause` and the `lp_cap_asym_pause` provide the advertised capabilities of the link partners.

`lp_cap_pause` When `lp_cap_pause` is 1, the link-partner can both assert and respond to flow control. This is the pre-Gigabit, symmetric mode of operation, and implies a full (both send and receive) implementation of the PAUSE mechanism within the device. In addition, if `lp_cap_asym_pause` is 1, the link-partner can operate either symmetrically or asymmetrically in either direction.

If `lp_cap_pause` is 0, the flow-control behavior supported by the link-partner is determined by `lp_cap_asym_pause`. If the value of `lp_cap_asym_pause` is 1, the link-partner can assert flow control, but cannot respond to any pause-frames sent to it.

No flow control is available when both `lp_cap_pause` and `lp_cap_asym_pause` are 0.

`lp_cap_asym_pause` Asymmetric pause capability

When `adv_*pause_cap` and `lp_*pause_cap` are compared on completion of auto-negotiation, the chosen flow control mechanism for the link depends on what is most meaningful.

`link_tx_pause` Link partner can assert flow control by sending pause frames when congestion is experienced.

`link_rx_pause` Link partner can respond to pause frames received.

The following truth table illustrates the meaningful flow control combinations related to local and link partner configurations. The abbreviations below are used in the table.

AA `adv_cap_asym_pause`

AP `adv_cap_pause`

LAC `lp_cap_asym_pause`

LPC `lp_cap_pause`

LA `link_asym_pause`

LP `link_pause`

AA	AP	LAC	LPC	LA	LP	Description
1	0	1	1	1	0	Local station will Tx a pause when Rx is congested.
0	1	0	1	0	1	Flow control in both Rx and Tx directions.
x	1	1	0	1	1	Local station honors received Pause frames by temporarily suspending Transmit.
x	x	x	x	0	0	All other combinations: Flow control not available on the link

When forced mode is enabled, the current setting of `adv_cap_asym_pause` and `adv_cap_pause` are used for the link. The `link_asym_pause` and `link_pause` become equal to the current `adv_cap_asym_pause` and `adv_cap_pause` settings. The above table also applies in forced mode, but the link partner configuration must be checked to verify that flow control is operating on the link.

関連項目 `dladm`(1M), `driver.conf`(4), `bge`(7D), `dlpi`(7P), `eri`(7D), `gld`(7D), `hme`(7D), `qfe`(7d)

注意事項 When `adv_cap_autoneg` is set to `0`, the highest priority speed and duplex is used for forced mode.

The highest priority is the highest speed at full duplex. The lowest priority is the lowest speed at half duplex.

MII transceivers can exist internally to a system or can be connected to an external MII connector. Typically, an internal transceiver has an `xcvr_addr` of `1`, while an external connection has an `xcvr_addr` of `0`.

名前

ipfilter – IP パケットフィルタ処理ソフトウェア

機能説明

IP フィルタは、Solaris システムにパケットフィルタ処理機能を提供するソフトウェアです。適切に設定されたシステムでは、ファイアウォールを構築するために使用できます。

Solaris IP フィルタは、Solaris オペレーティングシステムとともにインストールされます。ただし、パケットフィルタ処理はデフォルトでは有効になっていません。IP フィルタ機能を有効にしてアクティブ化する手順については、ipf(1M) を参照してください。

場所プロファイルとの相互作用

IP フィルタの構成およびアクティブ化は場所プロファイルで管理されます (場所プロファイルの詳細については、netcfg(1M) を参照)。これらのプロファイルは、固定 (つまり、ネットワーク構成は従来の方法で管理される) とリアクティブ (つまり、ネットワーク構成は自動的に管理され、プロファイルに指定されたポリシールールに従ってネットワーク環境の変更に反応する) のいずれかです。

固定の場所 (現在は、DefaultFixed 場所の 1 つのみ) がアクティブな場合は、場所を無効にしたときに SMF リポジトリへの変更がその場所に適用され、場所を再度有効にしたときに復元されます。

リアクティブな場所がアクティブのときは、変更は SMF リポジトリに直接適用すべきではありません。これらの変更は場所プロファイルに保持されないため、場所が無効にされるか、svc:/network/physical:default および svc:/network/location:default によって管理されるシステムのネットワーク構成がリフレッシュまたは再起動された場合に失われます。代わりに、変更は、netcfg コマンドを使用して場所自体に適用すべきです。これによって、変更は場所プロファイルリポジトリに保存され、SMF リポジトリにも適用されます (現在アクティブな場所に対して変更が行われた場合)。

IPv4 フィルタ構成ファイルが ipfilter-config-file プロパティーで指定されている場合は、ipfilter SMF サービスが有効になります。追加のフィルタタイプを有効にするために、ipfilter-v6-config-file、ipnat-config-file、および ippool-config-file プロパティーを指定することも可能です。

サービス

ipfilter SMF サービスは、start、stop、restart、および refresh メソッドをサポートします。これらのメソッドは、svcadm(1M) を使用して呼び出されます。

start ipfilter カーネルモジュールをロードし、構成に従ってファイアウォールまたは NAT 規則をアクティブにします。

stop 適用されたすべてのファイアウォールと NAT 規則、および作成されたアクティブなセッション情報をクリアします。ネットワーク接続が有効な状態でのサービスの停止は、ネットワークトラフィックがホストに入るリスクがない場合にのみ実行するようにしてください。

restart ipfilter サービスの停止と開始を実行します。動作中のファイアウォールに対してこのメソッドを使用すると、トラフィックがファイア

ウォールに侵入したり、フィルタリングされずに通過することができる、危険にさらされやすい期間が生じることになります。

`refresh` 現在の構成をロードし、セキュリティーポリシーが両方ともアクティブに使用されない時間をまったく生じさせずに古い構成から新しい構成に切り替えます。

ホストベースのファイアウォール

IP フィルタ構成の管理を簡素化するためにファイアウォールフレームワークが作成され、ユーザーはシステムレベルとサービスレベルのファイアウォールポリシーを記述することによって IP フィルタを構成できます。ユーザーが定義したファイアウォールポリシーから、必要なシステム動作が得られるようにフレームワークによって一連の IP フィルタ規則が生成されます。ユーザーは、特定のホスト、サブネット、およびインタフェースからのネットワークトラフィックを許可または拒否するよう、システムおよびサービスのファイアウォールポリシーを指定します。これらのポリシーが一連のアクティブな IPF 規則に変換されることにより、指定されたファイアウォールポリシーが適用されます。

ユーザーはフレームワークを利用しないことを選択した場合でも、独自の `ipf` 規則ファイルを指定できます。カスタマイズされた規則を有効にする方法については、`ipf(1M)` を、`ipf` 規則の構文を見つけるには、`ipf(4)` を参照してください。

IPF では、`ipmon(1M)` サービスを使用してファイアウォールイベントのログをとります。`ipmon` SMF サービスは、`ipfilter` SMF サービスに依存しています。`ipfilter` が “`svcadm enable ipfilter`” によって自動的に有効になるとすぐに、`ipmon` サービスが `ipfilter` サービス開始メソッドによって一時的に有効になります。

モデル

このセクションでは、ホストベースのファイアウォールフレームワークについて説明します。ファイアウォールポリシーを構成する方法については、`svc.ipfd(1M)` を参照してください。

ユーザーは、優先度の異なる 3 つのレイヤーを利用して、必要な動作を実現できます。

Global Default
: Global Default - システム全体のデフォルトファイアウォールポリシー。このポリシーは、サービスで独自にファイアウォールポリシーが変更されている場合を除き、すべてのサービスによって自動的に継承されます。

Network Services
: Global Default よりも優先されます。サービスのポリシーは、Global Default ポリシーにかかわらず、その特定のポートに対するトラフィックを許可または禁止します。

Global Override
: もう一つのシステム全体のポリシーで、Network Services レイヤーの個々のサービスの要求よりも優先されます。

```
Global Override
      |
      |
Network Services
      |
      |
Global Default
```

ファイアウォールポリシーには、ファイアウォールモードと、必要に応じてネットワークソースを設定します。ネットワークソースとは、システムが着信トラフィックを受け取る可能性のある、発信元の IP アドレス、サブネット、およびローカルネットワークインタフェースです。基本的なファイアウォールモードは次のとおりです。

None
ファイアウォールなしで、すべての着信トラフィックを許可します。

Deny
すべての着信トラフィックを許可しますが、指定した発信元からの着信トラフィックは拒否します。

Allow
すべての着信トラフィックを拒否しますが、指定した発信元からの着信トラフィックは許可します。

レイヤーの詳細

最初のシステム全体のレイヤー Global Default では、任意の着信トラフィックに適用するファイアウォールポリシーを定義します。たとえば、ある IP アドレスからのトラフィックをすべて許可またはブロックするように指定します。これにより、すべての着信トラフィックをブロックする、または望まない発信元からの着信トラフィックをすべてブロックするポリシーを簡単に作成できます。

Network Services レイヤーには、`telnetd`、`sshd`、`httpd` などのリモートクライアントにサービスを提供するローカルプログラムのためのファイアウォールポリシーが含まれています。このようなプログラム (ネットワークサービス) にはそれぞれ、サービスへのアクセスを制御する独自のファイアウォールポリシーがあります。最初は、サービスのポリシーは Global Default ポリシーを継承するように設定されています。つまり、「グローバルデフォルトを使用する」モードです。これにより、1 つのポリシーを Global Default レイヤーで設定し、それをすべてのサービスで簡単に継承することができます。

サービスのポリシーが Global Default ポリシーと異なっている場合は、サービスのポリシーが優先されます。あるサブネットからのトラフィックをすべてブロックするように Global Default ポリシーが設定されている場合でも、SSH サービスではそのサブネットの特定のホストからはアクセスを許可するように構成できます。Network Service レイヤーは、すべてのネットワークサービスに関するすべてのポリシーから成ります。

2 番目のシステム全体のレイヤー Global Override にも、任意の着信ネットワークトラフィックに適用されるファイアウォールポリシーが含まれています。このポリシーはもっとも優先度が高く、ほかのレイヤーのポリシーよりも優先されるため、個々のネットワークサービスの要求もオーバーライドします。たとえば、サービスのポリシーにかかわらず、悪意のある既知の発信元をブロックする場合に適しています。

ユーザーとの対話

このフレームワークは IP フィルタ機能を活用するため、`svc:/network/ipfilter` が有効になっている場合だけアクティブになり、`network/ipfilter` が無効になっている場合はアクティブになりません。同様に、ネットワークサービスのファイアウォールポリシーは、サービスが有効になっている場合だけアクティブになり、サービスが無効になっている場合はアクティブになりません。ファイアウォールがアクティブになっているシステムには、実行中または有効になっている各ネットワークサービスの IP フィルタ規則が存在し、ファイアウォールモード `None` ではないシステム全体のポリシーが存在します。

ユーザーは、システム全体のポリシーと各ネットワークサービスのポリシーを設定することにより、ファイアウォールを構成します。ファイアウォールポリシーの構成方法については、svc.ipfd(1M) を参照してください。

ファイアウォールフレームワークは、ポリシーの構成と、そのポリシーから IP フィルタ規則を生成して適用することによって必要な IP フィルタ構成を実現するメカニズムから成ります。設計およびユーザーとの対話のサマリーは次のとおりです。

- システム全体のポリシーは `network/ipfilter` に保存されます
- ネットワークサービスのポリシーは各 SMF サービスに保存されます
- ユーザーは `network/ipfilter` を有効にすることでファイアウォールをアクティブにします (`ipf(1M)` を参照)
- ユーザーはネットワークサービスを有効または無効にすることで、そのサービスのファイアウォールをアクティブまたは非アクティブにします
- システム全体またはサービスごとのファイアウォールポリシーを変更すると、システムのファイアウォール規則が更新されます

属性

属性についての詳細は、`attributes(5)` を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実

関連項目

`svcs(1)`、`ipf(1M)`、`ipmon(1M)`、`ipnat(1M)`、`netcfg(1M)`、`svcadm(1M)`、`svc.ipfd(1M)`、`ipf(4`

『Solaris のシステム管理 (IP サービス)』

注意事項

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

```
svc:/network/ipfilter:default
```

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

IP フィルタの起動時構成ファイルは `/etc/ipf` に保存されます。

名前

isalist – Solaris ソフトウェアに認識されるネイティブな命令セット

機能説明

isalist(1) によって、および sysinfo(2) の SI_ISALIST コマンドによって返される可能性のある命令セットの名前を、ここに一覧表示します。

このリストでは、命令セットファミリ別に、おおむね後ろに行くほど高速になるように名前が並んでいます。isalist(1) および sysinfo(2) の表示とは逆の順序です。次の値リストでは、通し番号はおおむねパフォーマンスの向上を表し、アルファベットの付いた項目は、相互排他的関係にあるか順序付け不可能なものです。

この機能は廃止されており、Solaris の将来のバージョンでは削除されることがあります。次のリストは、最近のプロセッサに追加されているすべての拡張機能を反映しているわけではありません。命令セット拡張機能のより適切な処理方法については、getisax(2) を参照してください。

SPARC プラットフォーム

適切と思われる箇所では、Sun の C 4.0 コンパイラの -xarch オプションの指定した値と対応する値が示されます。ほかのコンパイラにも同様のオプションがあります。

1a. sparc
: 『SPARC Architecture Manual, Version 8』(Prentice-Hall, Inc. 発行、1992 年) で定義された SPARC V8 命令セットを示します。特定のシステムでは、一部の命令 (整数の乗除、FSMULD および quad オペランドでのすべての浮動小数点演算) がカーネルによりエミュレートされます。

1b. sparcv7
: sparc の場合と同じです。これは、Sun の C 4.0 コンパイラの –xarch=v7 オプションで生成されるコードに対応します。

2. sparcv8–fsmuld
: sparc と同様ですが、整数の乗除がハードウェアで実行される必要がある点が異なります。これは、Sun の C 4.0 コンパイラの –xarch=v8a オプションで生成されるコードに対応します。

3. sparcv8
: sparcv8–fsmuld と同様ですが、FSMULD がハードウェアで実行される必要がある点が異なります。これは、Sun の C 4.0 コンパイラの –xarch=v8 オプションで生成されるコードに対応します。

4. sparcv8plus
: SPARC V8 命令セットに加え、『The SPARC Architecture Manual, Version 9』(Prentice-Hall, Inc. 発行、1994 年) で定義されている SPARC V9 命令セット内の命令 (『The V8+ Technical Specification』に従って使用可能) も示します。これは、Sun の C 4.0 コンパイラの –xarch=v8plus オプションで生成されるコードに対応します。

	5a. sparcv8plus+vis	sparcv8plus と同様ですが、さらに『The V8+ Technical Specification』に従って使用可能な UltraSPARC I 視覚化命令 (VIS) が加わります。これは、Sun の C 4.0 コンパイラの –xarch=v8plusa オプションで生成されるコードに対応します。
	5b. sparcv8plus+fmuladd	sparcv8plus と同様ですが、さらに富士通 SPARC64 の Floating Multiply-Add 命令および Multiply-Subtract 命令が加わります。
	6. sparcv9	『SPARC Architecture Manual, Version 9』(Prentice-Hall, Inc. 発行、1994 年) で定義された SPARC V9 命令セットを示します。
	7a. sparcv9+vis	sparcv9 と同様ですが、さらに UltraSPARC I 視覚化命令 (VIS) が加わります。
	7b. sparcv9+vis2	sparcv9 と同様ですが、さらに UltraSPARC III 視覚化命令 (VIS) が加わります。
	7c. sparcv9+fmuladd	sparcv9 と同様ですが、さらに富士通 SPARC64 の Floating Multiply-Add 命令および Multiply-Subtract 命令が加わります。
x86 プラットフォーム	1.i386	『The i386 Microprocessor Programmer's Reference Manual』で説明されている Intel 80386 命令セット。
	2. i486	『The i486 Microprocessor Programmer's Reference Manual』で説明されている Intel 80486 命令セット。(これは、事実上 i386 に加え、CMPXCHG、BSWAP、および XADD 命令。)
	3. pentium	『The Pentium Processor User's Manual』で説明されている Intel Pentium の命令セット。(これは、事実上 i486 に加え、CPU_ID 命令および CPU_ID 命令が提示する機能すべてが存在。)
	4. pentium+mmx	pentium と同様ですが、MMX 命令の存在が保証されています。
	5. pentium_pro	『The PentiumPro Family Developer's Manual』で説明されている Intel PentiumPro の命令セット。(これは、事実上 pentium に加え、CMOVcc、FCMOVcc、FCOMI、および RDPMC 命令の存在が保証されています。)
	6. pentium_pro+mmx	pentium_pro と同様ですが、MMX 命令の存在が保証されています。
	7. amd64	『AMD64 Architecture Programmer's Manual』で説明されている AMD Opteron 命令セット。

関連項目 isalist(1), getisax(2), sysinfo(2)

名前

kerberos – Solaris Kerberos 実装の概要

機能説明

Solaris Kerberos の実装 (以降、「Kerberos」と短縮する場合もあり) によって、ネットワーク環境内のクライアントが認証されるため、セキュアなトランザクションが可能になります。(クライアントはユーザーまたはネットワークサービスです。)Kerberos では、クライアントの同一性および転送されたデータの信頼性が検証されます。Kerberos は「シングルサインオン」システムです。つまり、ユーザーはセッションの開始時にのみ、パスワードを入力する必要があります。Solaris Kerberos は、MIT で開発された Kerberos(TM) システムに基づいて実装され、異機種混在ネットワーク上で Kerberos V5 システムとの互換性があります。

Kerberos は、クライアントを一意に識別し、有効期間に限りがあるチケットをクライアントに付与することによって動作します。チケットを所有するクライアントは、権限が付与されているネットワークサービスについて自動的に検証されます。たとえば、有効な Kerberos チケットを持つユーザーは、自分自身の身元を証明しなくても、Kerberos が動作している別のマシンに rlogin できます。各クライアントは一意のチケットを持っているため、身元が保証されます。

チケットを入手するには、まずクライアントは `kinit(1)` コマンドまたは PAM モジュールを使用して、Kerberos セッションを初期化する必要があります。(`pam_krb5(5)` を参照)。`kinit` によってパスワードを求めるプロンプトが表示され、*Key Distribution Center* (KDC) との通信が行なわれます。KDC によって、チケット認可チケット (TGT) が返され、パスワードの確認を求めるプロンプトが表示されます。クライアントがパスワードを確認すると、チケット認可チケットを使用して、特定のネットワークサービスのチケットを取得できます。チケットは透過的に付与されるため、ユーザーが管理について心配する必要はありません。`klist(1)` コマンドを使用すると、現在のチケットを表示できます。

チケットは、インストール時に設定されたシステムポリシーに従って有効になります。たとえば、チケットには有効なデフォルトの有効期間があります。root に属するチケットなどの特権チケットの有効期間が非常に短くなるように、ポリシーに追加指示することもできます。ポリシーでは、いくつかのデフォルトルールを上書きできます。たとえば、クライアントは、デフォルトよりも有効期間が長いまたは短いチケットをリクエストできます。

`kinit` を使用すると、チケットを更新できます。チケットは転送可能でもあるため、あるマシン上で付与されたチケットを別のホスト上で使用できます。`kdestroy(1)` を使用すると、チケットを破棄できます。`.logout` ファイルに `kdestroy` への呼び出しを含めることをお勧めします。

Kerberos では、クライアントは主体と呼ばれます。主体の形式は次のとおりです:

```
primary/instance@REALM
```

プライマリノード　　ユーザー、ホスト、またはサービス。

インスタンス　プライマリの資格です。プライマリがキーワード `host` で指定されたホストの場合、インスタンスはそのホストの完全指定ドメイン名です。プライマリがユーザーまたはサービスの場合、インスタンスはオプションです。`admin` や `root` などの一部のインスタンスは特権です。

レルム　Kerberos で、ドメインに相当するものです。実際に、ほとんどの場合でレルムは DNS ドメイン名に直接マップされています。Kerberos レルムでは大文字のみが使用されます。主体名の例については、「使用例」を参照してください。

Kerberos では、ユーザー認証以外にも、General Security Services API (GSS-API) を利用した 2 種類のセキュリティーサービスが提供されます: 転送されたデータの有効性を認証する *integrity* と、転送されたデータを暗号化する *privacy*。開発者は RPCSEC_GSS API インタフェースを使用することによって、GSS-API を利用できます (`rpcsec_gss(3NSL)` を参照)。

使用例

例 1　有効な主体名の例

次に、有効な主体名の例を示します:

```
joe
joe/admin
joe@ENG.ACME.COM
joe/admin@ENG.ACME.COM
rlogin/bigmachine.eng.acme.com@ENG.ACME.COM
host/bigmachine.eng.acme.com@ENG.ACME.COM
```

最初の 4 つのケースはユーザー主体です。最初の 2 つのケースでは、ユーザー `joe` がクライアントと同じレルム内にあるため、レルムが指定されていないことが想定されます。`joe` と `joe/admin` は、同じユーザーに使用されている場合でも、別々の主体であることに注意してください。`joe/admin` は、`joe` とは別の特権を持っています。5 番目のケースは サービス主体、最後のケースはホスト主体です。ホスト主体には、`host` という語が必要です。ホスト主体では、インスタンスは完全指定ホスト名です。`admin` および `host` という語は予約済みのキーワードであることに注意してください。

関連項目

`kdestroy(1)`、`kinit(1)`、`klist(1)`、`kpasswd(1)`、`krb5.conf(4)`、`krb5envvar(5)`

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

注意事項

以前のリリースの Solaris オペレーティングシステムでは、Solaris Kerberos の実装は「SEAM (Sun Enterprise Authentication Mechanism)」と呼ばれていました。

ユーザー名を入力し、`kinit` が次のメッセージを返した場合:

```
Principal unknown (kerberos)
```

Kerberos ユーザーとして登録されていません。システム管理者に問い合わせるか、『Oracle Solaris 11.1 の管理: セキュリティーサービス』を参照してください。

名前

krb5_auth_rules – Kerberos V5 承認の概要

機能説明

Kerberos 化されたバージョンの `ftp`、`rcp`、`rlogin`、`rsh`、`ssh`、`telnet`、または `SSH` クライアントを使用してサーバーに接続すると、元のユーザーの識別情報は Kerberos V5 認証システムに対する認証を受ける必要があります。その後、`~/.k5login` ファイルと `gsscred` テーブルに適切なエントリが存在する場合や、デフォルトの GSS/Kerberos 認証ルールが正常に Kerberos 主体名を UNIX ログイン名にマップしている場合、アカウントアクセスの認証が可能になります。

セキュリティーの問題を回避するには、クライアントがアクセスを試みているサーバー上のリモートユーザーが `~/.k5login` ファイルを所有している必要があります。ファイルには、*principal/instance@realm* 形式の Kerberos 主体名で構成される非公開の承認リストが含まれている必要があります。Kerberos 主体名の */instance* 変数はオプションです。たとえば、`jdb@ENG.ACME.COM` や `jdb/happy.eng.acme.com@ENG.ACME.COM` などのさまざまな主体名は同等ではありませんが、正当な Kerberos 主体です。`~/.k5login` ファイルがリモートユーザーアカウントのログインディレクトリに配置され、元のユーザーがファイルに指定された主体のいずれかに対する認証を受けることができる場合、クライアントにアクセス権が付与されます。Kerberos 主体名の詳細については、`gkadmin(1M)` および `kadm5.acl(4)` を参照してください。

`~/.k5login` ファイルがリモートユーザーのログインアカウントに見つからなかった場合は、元のユーザーに関連付けられた Kerberos V5 主体名が `gsscred` テーブルと比較してチェックされます。`gsscred` テーブルが存在し、そのテーブルで主体名が一致している場合、テーブルにリストされている Unix ユーザー ID がクライアントがアクセスを試みているユーザーアカウントに対応していれば、アクセス権が付与されます。Unix ユーザー ID が一致しない場合、アクセスは拒否されます。`gsscred(1M)` を参照してください。

たとえば、`gsscred` テーブルにリストされている元のユーザーの主体名が `jdb@ENG.ACME.COM`、`uid` が `23154` の場合、`23154` がユーザーアカウントデータベースにリストされている `jdb-user` の `uid` でもあれば、`jdb-user` アカウントへのアクセス権が付与されます。`passwd(4)` を参照してください。

最後に、`~/.k5login` ファイルが存在せず、元のユーザーの Kerberos V5 識別情報が `gsscred` テーブルに存在しない場合や、`gsscred` テーブル自体が存在しない場合、次の条件 (デフォルトの GSS/Kerberos 認証ルール) を満たせば、クライアントにアカウントへのアクセス権が付与されます:

- 認証された主体名のユーザー部分がクライアントが指定した Unix アカウント名と同じである。
- `krb5.conf(4)` *auth_to_local_realm* パラメータを使用してレルムを作成しない場合を除いて、クライアントとサーバーのレルム部分は同じである。
- Unix アカウント名がサーバーに存在する。

たとえば、元のユーザーの主体名が jdb@ENG.ACME.COM で、サーバーのレルムが SALES.ACME.COM の場合、jdb がサーバー上の有効なアカウント名であっても、クライアントのアクセスは拒否されます。これは、レルム SALES.ACME.COM と ENG.ACME.COM が異なるためです。

krb5.conf(4) *auth_to_local_realm* パラメータによっても承認は影響を受けます。デフォルト以外のレルムは、認証済みの name-to-local name マッピングのデフォルトレルムと同等とみなすことができます。

ファイル

~/.k5login　ユーザーアカウントごとの認証ファイルです。

/etc/passwd　システムアカウントファイルです。この情報はディレクトリサービスに存在することもあります。passwd(4) を参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実

関連項目

ftp(1)、rcp(1)、rsh(1)、telnet(1)、gkadmin(1M)、gsscred(1M)、kadm5.acl(4)、krb5.conf(4)、pa

名前

krb5envvar – Kerberos 環境変数

機能説明

Kerberos メカニズムには、アプリケーションのニーズを満たすために、さまざまな動作を構成する多数の環境変数が用意されています。Kerberos 機構では、次の環境変数が使用されます:

KRB5_KTNAME
鍵テーブルファイルの場所を指定するためのメカニズムで使用されます。この変数には、次の値を設定できます:

[[<*kt type*>:]<*file name*>]

ここで、<*kt type*> には FILE または WRFILE を指定できます。FILE は読み取り操作用、WRFILE は書き込み操作用です。<*file name*> は、keytab ファイルの場所です。

r

KRB5_KTNAME が定義されない場合のデフォルト値は次のとおりです:

FILE:/etc/krb5/krb5.keytab

keytab ファイルは、永続的に資格を格納するために使用され、一般にサービスデーモンで使用されます。

FILE タイプを指定することは、関連するファイル上の後続操作が呼び出しプロセスで読み取り可能であることが前提となっています。暗号化されていない鍵を取得する必要のある主体のセットでのみファイルが読み取り可能になるように注意してください。

WRFILE タイプは、kadmin(1M) コマンドで使用されます。このタイプを指定すると、管理者は追加のコマンド行引数を使用してファイルの場所を指定しなくても、書き込む代替の keytab ファイルを指定できます。

KRB5CCNAME
資格キャッシュの場所を指定するためのメカニズムで使用されます。この変数には、次の値を設定できます:

[[<*cc type*>:]<*file name*>]

ここで、<*cc type*> には FILE または MEMORY を指定できます。<*file name*> は主体の資格キャッシュの場所です。

KRB5CCNAME が定義されない場合のデフォルト値は次のとおりです:

FILE:/tmp/krb5cc_<*uid*>

ここで、<*uid*> はキャッシュファイルを作成したプロセスのユーザー ID です。

資格キャッシュファイルは、主体に付与されているチケットを格納するために使用されます。

FILE タイプを指定することは、関連するファイル上の後続操作が呼び出しプロセスで読み書き可能であることが前提となっています。資格にアクセスする必要のある主体のセットのみがファイルにアクセス可能になるように注意してください。ほかのユーザーが書き込みアクセス権を持っているディレクトリ内に資格ファイルがある場合、そのディレクトリのスティッキービットを設定する必要があります (chmod(1) を参照)。

MEMORY タイプの資格キャッシュは、呼び出しプロセスの存続期間中に一時キャッシュを作成する場合などの特殊なケースでしか使用されません。

KRB5RCNAME
リプレイキャッシュのタイプと場所を指定するためのメカニズムで使用されます。この変数には、次の値を設定できます:

[[<*rc type*>:]<*file name*>]

ここで、<*rc type*> には FILE、MEMORY、または NONE を指定できます。<*file name*> が関連するのは、リプレイキャッシュファイルタイプを指定する場合のみです。

定義されない場合のデフォルト値は次のとおりです:

FILE:/var/krb5/rcache/root/rc_<*service*>

root がプロセスを所有している場合、または:

FILE:/var/krb5/rcache/rc_<*service*>

root 以外のユーザーがプロセスを所有している場合。<*service*> は、リプレイキャッシュファイルに関連付けられたサービスプロセス名です。

リプレイキャッシュは、Kerberos で認証データのリプレイを検出する際に使用されます。これにより、ネットワーク上の認証メッセージを取得するユーザーは、これらのメッセージを再送信しても、サーバーに対する認証を行なうことができません。

FILE タイプのリプレイキャッシュを指定する場合は、別のユーザーがリプレイキャッシュファイルを削除できないように注意してください。リプレイキャッシュの所有者のみがリプレイキャッシュパスのすべてのディレクトリに書き込み可能であるか、またはその他のユーザーが書き込み権を持っているリプレイキャッシュパスのすべてのディレクトリにスティッキービット (“ t”) が設定されていることを確認します。

MEMORY タイプのリプレイキャッシュを指定する場合は、非永続性キャッシュを使用することで発生するわずかなセキュリティーリスクに対するパフォーマンスのトレードオフを比較検討する必要があります。次の状況になると、システムのリブート中にリスクが発生します:

- リブート前に最後にリプレイキャッシュに書き込んでから、Kerberos 化されたサーバーアプリケーションが実行されている時点までの期間は、Kerberos クロックスキューよりも短くなります (krb5.conf(4) を参照)。

NONE リプレイキャッシュ時間を指定する場合は、これによりリプレイキャッシュが無効になることと、発生するすべてのセキュリティーリスクについて理解する必要があります。これには、マニュアルページの該当セクションで説明されたすべてのリスクが含まれます。

このような状況では、サーバーアプリケーションは、Kerberos 認証データのリプレイを (最後の書き込み時間とクロックスキューとの差異まで) 許容できます。一般に、これは短い猶予時間です。接続の受け入れを開始するまでの時間がクロックスキューよりもサーバーアプリケーションの方が長い場合は、リプレイのリスクはありません。

前述のリスクは、/var/run や /tmp などのスワップファイルシステム上にリプレイキャッシュが存在するときに、FILE タイプのリプレイキャッシュを使用する場合と同じです。

MEMORY タイプのリプレイキャッシュでは、ディスク I/O が存在しないことによって、FILE タイプよりもパフォーマンスが改善されています。このことは、FILE リプレイキャッシュがスワップ (/tmp や /var/run など) のメモリーに基づくファイルシステムに存在する場合でも該当します。

MEMORY タイプのキャッシュはプロセスごとのキャッシュであるため、これらのタイプのキャッシュを使用する際には十分に考慮する必要があります。MEMORY タイプのキャッシュで問題が発生する可能性がある一例として、セキュリティーコンテキストを確立するために複数のプロセスがアプリケーションで使用されている場合があります。このような場合、リプレイキャッシュメモリーはプロセス間で共有されないため、リプレイ攻撃の可能性があります。

KRB5_CONFIG
KRB5_CONFIG で指定された別のファイルから構成パラメータを Kerberos ライブラリコードで読み取ることができるように、/etc/krb5/krb5.conf ファイルのデフォルトの場所を変更できます。ksh(1) から kinit を使用する例:

```
KRB5_CONFIG=/var/tmp/krb5.conf kinit
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/security/kerberos-5
インタフェースの安定性	不確実

関連項目

chmod(1)、kinit(1)、klist(1)、ksh(1)、kadmin(1M)、kadmind(1M)、krb5.conf(4)、attribute

名前

kssl, KSSL – カーネル SSL プロキシ

機能説明

KSSL は、SSL/TLS プロトコルに対して透過的なサーバー側プロキシです。SSL トラフィックはカーネルで処理されるため、コンテキストの切り替えが回避され、Oracle Solaris 暗号化フレームワークのカーネルプロバイダに直接アクセスすることによって、パフォーマンスが改善されます。KSSL を使用すると、TCP 経由で平文でのみ通信可能なアプリケーションも SSL で保護できます。

KSSL はカーネルで構成され、アプリケーションからの平文データを渡したり、受け取ったりします。1 台の SSL サーバーとしてまとめてクライアントに表示されます。

KSSL が構成されているサーバー側アプリケーションは、事前に SSL で保護されたデータを受信していることに気付きません。KSSL は 1 つのポート (443 など) で SSL トラフィックを受信し、処理を実行して、別のポート (8080 など) で待機しているアプリケーションに平文データを渡します。送信方向の場合も同様に、アプリケーションが平文データを送信し、KSSL が SSL レコードを生成して、クライアントに送信します。したがって、アプリケーションには SSL の設定は必要ありません。

システム上に複数の KSSL インスタンスを構成でき、ポート、証明書、鍵、暗号化方式などそれぞれ個別のプロパティーセットがあります。ksslcfg(1M) を参照してください。カーネル内の各 KSSL インスタンスは SMF サービスとして追跡されます。smf(5) を参照してください。

KSSL では、TCP over IPv4 と TCP over IPv6 の両方で渡されたレコードの SSL 処理が行なわれます。

KSSL では、次のプロトコルがサポートされています: SSLv3、TLSv1.0

関連項目

ksslcfg(1M)、smf(5)

T.Dierks, C.Allen, RFC 2246, The TLS Protocol Version 1.0, The Internet Society, 1999.

名前

labels – Solaris Trusted Extensions ラベル属性

機能説明

ラベルは必須ポリシーの決定に使用される属性です。ラベルは、すべてのサブジェクト (通常はプロセス) およびサブジェクトへとアクセス可能なオブジェクト (通常はファイルのようにデータを保持するもの) に、明示的または暗黙的に関連付けられます。デフォルトの Trusted Extensions 必須ポリシーラベルは、サイトのセキュリティー管理者により、label_encodings(4) 内で定義されます。

必須ポリシー

Solaris Trusted Extensions の有効期間中に、さまざまな必須ポリシーを提供できます。

Trusted Extensions のデフォルト必須ポリシーは必須アクセス制御 (MAC) ポリシー (Lattice の Bell-LaPadula Model の必須ポリシー、Simple Security Property、および *-Property (Star Property) と同等) で、制限された上位書き込みを持ちます。デフォルト必須ポリシーは、Non-Inteference の Goguen and Mesegeur モデルとも同等です。

この MAC ポリシーの場合、2 つのラベル admin_low および admin_high が常に定義されます。サイトのセキュリティー管理者は、label_encodings(4) 内でその他のラベルすべてを定義します。admin_low は、ユーザーが読み取り (表示) 可能な通常の Trusted Extensions オブジェクトすべてと関連付けられます。admin_high は、その他の Trusted Extensions オブジェクトすべてと関連付けられます。admin_high オブジェクトへの MAC 読み取り (表示) アクセス権を保持するのは、管理ユーザーだけです。また、admin_low オブジェクトや admin_high オブジェクトへの MAC 書き込み (変更) アクセス権を保持するのも管理ユーザーだけです。

人間が読み取り可能なラベル

ユーザーは、ラベルを文字列として対話的に処理します。グラフィカルユーザーインタフェースおよびコマンド行インタフェースには、label_encodings(4) の定義に従って文字列が表示されます。人間が読み取り可能なラベルは、それが表現するラベルで格付けされます。このため、ラベル A の文字列を読み取り可能 (表示可能、人間が読み取り可能な文字列と不透明な m_label_t 間の相互変換が可能) なのは、そのラベルへの読み取り (表示) アクセスを許可するラベルを持つサブジェクトだけです。

内部テキストラベル

公開アクセス可能な (admin_low) ネームサービスデータベースにラベルを格納する場合には、格付けされていない内部テキスト形式が使用されます。このテキスト形式は、この形式のラベルを作成した Trusted Extensions ソフトウェアリリースで提供されるインタフェース以外のインタフェースでの使用は意図されていません。

ラベルとアプリケーション

アプリケーションは、ラベルとの対話処理を行う際に、ラベルが不透明な (m_label_t) 構造体であると見なします。これら不透明な構造体のセマンティクスは、文字列から m_label_t への変換で定義されます。変換は、label_encodings(4) 内で定義されます。さまざまなアプリケーションプログラミングインタフェース (API) が、文字列と m_label_t 構造体間の変換を行います。さまざま API が、サブジェクト関連のラベルからオブジェクト関連のラベルへのアクセスをテストします。

属性

次の属性については、 attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	下記を参照。

ラベル実装は、label_encodings(4) の米国国防情報局 (DIA) MAC ポリシーを実装するシステムでは「確実」です。Trusted Extensions の将来のリリースにその他のポリシーが存在する可能性はありますが、それらは将来のリリースでは廃止されるか、label_encodings の補完的役割になる可能性があります。

内部テキストラベルはインタフェースではないため、Trusted Extensions のいずれかのリリースで変更される可能性があります。これらは、同一の Trusted Extensions ソフトウェアリリース上で入力および生成を行う場合だけを想定しています。

Trusted Solaris 8 アプリケーションでの移植性を高めるために、不透明な構造体名 bslabel_t、blevel_t、および bclear_t が、m_label_t と等価なものとして定義されています。これらのタイプは、m_label_t と同様に、不透明なポインタとして移植する必要があります。さまざまな Trusted Solaris 8 ラベルインタフェースに対して、同じ作業を行う必要があります。これらの Trusted Solaris 8 構造体およびインタフェースは廃止済みであり、Trusted Extensions の将来のリリースでは削除される可能性があります。

関連項目

chk_encodings(1M), blcompare(3TSOL), label_to_str(3TSOL), m_label_alloc(3TSOL), m_label_dup(3TSOL), m_label_free(3TSOL), str_to_label(3TSOL), label_encodings(4), attributes(5)

Bell, D.E.、および LaPadula, L. J. 共著『Secure Computer Systems: Unified Exposition and Multics Interpretation』、MTR-2997 Rev. 2、MITRE Corp., Bedford Mass.、1976 年 3 月。NTIS AD-A023 588/7。

Goguen, J.A.、および Mesegeur, J. 著:『Security Policies and Security Models』、Proceedings 1982 Symposium on Security and Privacy、IEEE Computer Society Press、1982 年、11-20 ページ。

Goguen, J.A.、および Mesegeur, J. 著:『Unwinding and Interference Control』、Proceedings 1984 Symposium on Security and Privacy、IEEE Computer Society Press、1984 年、75-86 ページ。

『Compartmented Mode Workstation Labeling: Encodings Format』

注意事項

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

名前 largefile – large file status of utilities

機能説明 A *large file* is a regular file whose size is greater than or equal to 2 Gbyte (2^{31} bytes). A *small file* is a regular file whose size is less than 2 Gbyte.

Large file aware utilities A utility is called *large file aware* if it can process large files in the same manner as it does small files. A utility that is large file aware is able to handle large files as input and generate as output large files that are being processed. The exception is where additional files are used as system configuration files or support files that can augment the processing. For example, the `file` utility supports the `-m` option for an alternative "magic" file and the `-f` option for a support file that can contain a list of file names. It is unspecified whether a utility that is large file aware will accept configuration or support files that are large files. If a large file aware utility does not accept configuration or support files that are large files, it will cause no data loss or corruption upon encountering such files and will return an appropriate error.

The following `/usr/bin` utilities are large file aware:

`adb`	`aliasadm`	`awk`	`bdiff`	`cat`
`chgrp`	`chmod`	`chown`	`cksum`	`cmp`
`compress`	`cp`	`csh`	`csplit`	`cut`
`dd`	`dircmp`	`du`	`egrep`	`fgrep`
`file`	`find`	`ftp`	`getconf`	`grep`
`gzip`	`head`	`join`	`jsh`	`ksh88`
`ksh`	`ln`	`ls`	`mailcompat`	`mailstats`
`mdb`	`mkdir`	`mkfifo`	`more`	`mv`
`nawk`	`page`	`paste`	`pathchck`	`pg`
`praliases`	`rcp`	`remsh`	`rksh88`	`rksh`
`rm`	`rmdir`	`rsh`	`sed`	`sh`
`sort`	`split`	`sum`	`tail`	`tar`
`tee`	`test`	`touch`	`tr`	`uncompress`
`uudcode`	`uuencode`	`vacation`	`wc`	`zcat`

The following `/usr/xpg4/bin` utilities are large file aware:

`awk`	`cp`	`chgrp`	`chown`	`du`
`egrep`	`fgrep`	`file`	`grep`	`ln`

```
ls              more            mv              rm              sed
sh              sort            tail            tr
```

The following `/usr/xpg6/bin` utilities are large file aware:

```
getconf         ls              tr
```

The following `/usr/sbin` utilities are large file aware:

```
editmap         install         makemap         mkfile          mknod
mvdir           swap
```

The following `/usr/lib` utilities are large file aware:

```
mail.local      sendmail        smrsh
```

See the USAGE section of the swap(1M) manual page for limitations of `swap` on block devices greater than 2 Gbyte on a 32–bit operating system.

The following `/usr/ucb` utilities are large file aware:

```
chown           from            ln              ls              sed
sum             touch
```

The `/usr/bin/cpio` and `/usr/bin/pax` utilities are large file aware, but cannot archive a file whose size exceeds 8 Gbyte – 1 byte.

The `/usr/bin/truss` utilities has been modified to read a dump file and display information relevant to large files, such as offsets.

nfs file systems

The following utilities are large file aware for `nfs` file systems:

```
/usr/lib/autofs/automountd              /usr/sbin/mount
/usr/lib/nfs/rquotad
```

ufs file systems

The following `/usr/bin` utility is large file aware for `ufs` file systems:

`df`

The following `/usr/lib/nfs` utility is large file aware for `ufs` file systems:

`rquotad`

The following `/usr/xpg4/bin` utility is large file aware for `ufs` file systems:

`df`

The following `/usr/sbin` utilities are large file aware for `ufs` file systems:

`clri`	`dcopy`	`edquota`	`ff`	`fsck`
`fsdb`	`fsirand`	`fstyp`	`labelit`	`lockfs`
`mkfs`	`mount`	`ncheck`	`newfs`	`quot`
`quota`	`quotacheck`	`quotaoff`	`quotaon`	`repquota`
`tunefs`	`ufsdump`	`ufsrestore`	`umount`	

Large file safe utilities

A utility is called *large file safe* if it causes no data loss or corruption when it encounters a large file. A utility that is large file safe is unable to process properly a large file, but returns an appropriate error.

The following `/usr/bin` utilities are large file safe:

`audioconvert`	`audioplay`	`audiorecord`	`comm`	`diff`
`diff3`	`diffmk`	`ed`	`lp`	`mail`
`mailcompat`	`mailstats`	`mailx`	`pack`	`pcat`
`red`	`rmail`	`sdiff`	`unpack`	`vi`
`view`				

The following `/usr/xpg4/bin` utilities are large file safe:

`ed`	`vi`	`view`

The following `/usr/xpg6/bin` utility is large file safe:

ed

The following /usr/sbin utilities are large file safe:

lpfilter lpforms

The following /usr/ucb utilities are large file safe:

Mail lpr

関連項目 lf64(5), lfcompile(5), lfcompile64(5)

名前

ldap – ネーミングリポジトリとしての LDAP

機能説明

LDAP (Lightweight Directory Access Protocol) は、ディレクトリサーバーにアクセスするための業界標準です。ldapclient(1M) を使用してクライアントを初期化し、ネームサービススイッチファイル /etc/nsswitch.conf でキーワード ldap を使用すると、Oracle Solaris クライアントは LDAP サーバーからネーミング情報を取得できます。ユーザー名、ホスト名、パスワードなどの情報は、DIT (Directory Information Tree) 内の LDAP サーバーに格納されます。DIT は、属性で構成されるエントリで構成されます。属性ごとに、1 つのタイプと 1 つ以上の値があります。

Oracle Solaris LDAP クライアントが LDAP サーバーからネーミング情報にアクセスするときは、LDAP v3 プロトコルを使用します。ネームサービスモデルが LDAP にマップされている RFC2307bis (ドラフト) で定義されたオブジェクトクラスおよび属性が、LDAP サーバーでサポートされている必要があります。RFC2307bis (ドラフト) で定義されたスキーマを使用する代わりに、その他のスキーマセットを使用するようにシステムを構成でき、スキーママッピング機能は 2 つのスキーマ間でマップするように構成されます。詳細については、『Oracle Solaris Administration: Naming and Directory Services』を参照してください。

ldapclient(1M) ユーティリティーで適切なディレクトリ、ファイル、および構成情報を設定すると、Oracle Solaris マシンを LDAP クライアントにすることができます。LDAP クライアントは、この構成情報をローカルキャッシュファイルに格納します。この構成情報にアクセスするには、ldap_cachemgr(1M) デーモンを使用します。また、このデーモンによって LDAP サーバーから構成ファイル内の情報がリフレッシュされるため、パフォーマンスおよびセキュリティーが改善されます。ネーミングサービスが正しく動作するには、常に ldap_cachemgr が実行される必要があります。

プロファイルを介して使用できる情報と、クライアントごとに構成された情報の 2 種類の構成情報があります。プロファイルには、クライアントがディレクトリにアクセスする方法に関する情報がすべて含まれます。プロキシユーザーの資格情報はクライアントごとに構成され、プロファイルを介してダウンロードされません。

プロファイルには、サーバーで目的の LDAP ドメインを検索するときにすべてのクライアントが必要とするサーバー固有のパラメータが含まれます。たとえば、この情報にはサーバーの IP アドレスおよび検索ベースの識別名 (DN) が含まれることがあります。これはクライアントの初期化時にクライアントでデフォルトプロファイルから構成され、有効期限が経過すると定期的に ldap_cachemgr デーモンによって更新されます。

クライアントプロファイルは LDAP サーバーに格納でき、ldapclient ユーティリティーで使用すれば LDAP クライアントを初期化できます。クライアントマシンを構成するもっとも簡単な方法は、クライアントプロファイルを使用することです。ldapclient(1M) を参照してください。

資格情報には、クライアントによって使用されるクライアント固有のパラメータが含まれています。この情報には、クライアントおよびパスワードのバインド DN (LDAP「ログイン」名) が含まれることがあります。これらのパラメータが必要な場合は、初期化時に ldapclient(1M) を使用して手動で定義されます。

ネーミング情報は、LDAP サーバー上のコンテナに格納されます。コンテナとは、ネームサービス情報を含む DIT 内のリーフ以外のエントリのことです。コンテナは NIS のマップに似ています。NIS データベースと LDAP 内のコンテナ間のデフォルトマッピングを次に示します。これらのコンテナの場所および名前は、serviceSearchDescriptors を使用することでオーバーライドできます。詳細については、ldapclient(1M) を参照してください。

データベース	オブジェクトクラス	コンテナ
passwd	posixAccount shadowAccount	ou=people,dc=...
group	posixGroup	ou=Group,dc=...
services	ipService	ou=Services,dc=...
protocols	ipProtocol	ou=Protocols,dc=...
rpc	oncRpc	ou=Rpc,dc=...
hosts ipnodes	ipHost ipHost	ou=Hosts,dc=... ou=Hosts,dc=...
ethers	ieee802Device	ou=Ethers,dc=...
bootparams	bootableDevice	ou=Ethers,dc=...
networks netmasks	ipNetwork ipNetwork	ou=Networks,dc=... ou=Networks,dc=...
netgroup	nisNetgroup	ou=Netgroup,dc=...
aliases	mailGroup	ou=Aliases,dc=...
publickey	nisKeyObject	
generic	nisObject	nisMapName=...,dc=...
printers	printerService	ou=Printers,dc=...
auth_attr	SolarisAuthAttr	ou=SolarisAuthAttr,dc=...
prof_attr	SolarisProfAttr	ou=SolarisProfAttr,dc=...
exec_attr	SolarisExecAttr	ou=SolarisProfAttr,dc=...

データベース	オブジェクトクラス	コンテナ
user_attr	SolarisUserAttr	ou=people,dc=...

クライアントのセキュリティーモデルは、使用される資格レベル、認証方法、および使用される PAM モジュールの組み合せによって定義されます。資格レベルでは、ディレクトリサーバーに対して認証を行なう際にクライアントが使用する必要のある資格が定義され、認証方式では選択する方法が定義されます。どちらにも複数の値を設定できます。Oracle Solaris LDAP の資格レベルでは、次の値がサポートされています:

```
anonymous
proxy
self
```

Oracle Solaris LDAP の認証方式では、次の値がサポートされています:

```
none
simple
sasl/CRAM-MD5
sasl/DIGEST-MD5
sasl/GSSAPI
tls:simple
tls:sasl/CRAM-MD5
tls:sasl/DIGEST-MD5
```

資格レベルが `self` として構成されている場合は、DNS が構成され、認証方式は `sasl/GSSAPI` である必要があります。DNS (たとえば、`hosts: dns files` および `ipnodes: dns files`) を使用するには、`/etc/nsswitch.conf` に `hosts` および `ipnodes` を構成する必要があります。

これらがディレクトリサーバーで構成されない場合は、`sasl/GSSAPI` で自動的に `GSSAPI` の機密性および整合性オプションが使用されます。

`self` の資格レベルでは、ユーザーごとのネームサービス検索、またはディレクトリサーバーへの接続時にユーザーの `GSSAPI` 資格を使用する検索が可能です。Kerberos V5 のこのモデルでは、現在、`GSSAPI` メカニズムのみがサポートされています。この資格レベルを使用する前に、Kerberos を構成する必要があります。詳細については、`kerberos(5)` を参照してください。

アクセス制御を使用すると、より強力な保護が提供され、サーバーで特定のコンテナまたはエントリに対するアクセス権を付与できるようになります。アクセス制御は、LDAP サーバーに定義および格納されているアクセス制御リスト (ACL) で指定されます。LDAP サーバーのアクセス制御リストは、SunOne Directory Server ではア

クセス制御命令 (ACI) と呼ばれます。各 ACI または ACL には、1 つ以上のディレクトリオブジェクトを指定します。たとえば、特定のコンテナの cn 属性、アクセス権を付与または拒否する 1 つ以上のクライアント、およびクライアントがオブジェクトに対して、またオブジェクトで実行できることを決定する 1 つ以上のアクセス権を指定します。クライアントにはユーザーまたはアプリケーションを指定できます。たとえば、読み取りおよび書き込みとしてアクセス権を指定できます。ネーミングリポジトリとして LDAP を使用する際の ACL および ACI の制限については、『Oracle Solaris Administration: Naming and Directory Services』を参照してください。

nsswitch.ldap と呼ばれる nsswitch.conf(4) ファイルのサンプルは、/etc ディレクトリで提供されています。ldapclient(1M) ユーティリティーを使用すると、これが /etc/nsswitch.conf にコピーされます。このファイルでは、nsswitch.conf ファイル内のさまざまなデータベースのリポジトリとして LDAP が使用されます。

次に、LDAP に関連するユーザーコマンドのリストを示します:

idsconfig(1M)	Solaris LDAP クライアントがサポート可能になるように、SunOne Directory Server の準備を行ないます。
ldapaddent(1M)	対応する /etc ファイルから LDAP エントリを作成します。
ldapclient(1M)	LDAP クライアントを初期化するか、ディレクトリに格納される構成プロファイルを生成します。
ldaplist(1)	LDAP ネーミング領域の内容をリストします。

ファイル

/var/ldap/ldap_client_cred /var/ldap/ldap_client_file	クライアントの LDAP 構成を含むファイル。これらのファイルを手動で変更しないでください。その内容は人間が読めるとは限りません。これを更新するには、ldapclient(1M) を使用します。
/etc/nsswitch.conf	ネームサービススイッチの構成ファイル。
/etc/nsswitch.ldap	LDAP およびファイルで構成されたネームサービススイッチのサンプル構成ファイル。
/etc/pam.conf	PAM フレームワーク構成ファイルです。
/etc/pam.d/*service*	代替の PAM フレームワーク構成ファイルです。

関連項目

ldaplist(1)、idsconfig(1M)、ldap_cachemgr(1M)、ldapaddent(1M)、ldapclient(1M)、nsswitch.

『Oracle Solaris Administration: Naming and Directory Services』

名前

lf64 – transitional interfaces for 64-bit file offsets

機能説明

The data types, interfaces, and macros described on this page provide explicit access to 64-bit file offsets. They are accessible through the transitional compilation environment described on the `lfcompile64(5)` manual page. The function prototype and semantics of a transitional interface are equivalent to those of the standard version of the call, except that relevant data types are 64-bit entities.

Data Types

The following tables list the standard data or struct types in the left-hand column and their corresponding explicit 64-bit file offset types in the right-hand column, grouped by header. The absence of an entry in the left-hand column indicates that there is no existing explicit 32-bit type that corresponds to the 64–bit type listed in the right—hand column. Note that in a 64-bit application, the standard definition is equivalent to the 64-bit file offset definition.

<aio.h>

struct aiocb	struct aiocb64
off_t aio_offset;	off64_t aio_offset;

<sys/dirent.h>

struct dirent	struct dirent64
ino_t d_ino;	ino64_t d_ino;
off_t d_off;	off64_t d_off;

<sys/fcntl.h>

struct flock	struct flock64
off_t l_start;	off64_t l_start;
off_t l_len;	off64_t l_len;
F_SETLK	F_SETLK64
F_SETLKW	F_SETLKW64
F_GETLK	F_GETLK64
F_FREESP	F_FREESP64
F_ALLOCSP	F_ALLOCSP64
	O_LARGEFILE

<sys/stdio.h>

fpos_t	fpos64_t

<sys/resource.h>

rlim_t	rlim64_t
struct rlimit	struct rlimit64
rlim_t rlim_cur;	rlim64_t rlim_cur;
rlim_t rlim_max;	rlim64_t rlim_max;
RLIM_INFINITY	RLIM64_INFINITY
RLIM_SAVED_MAX	RLIM64_SAVED_MAX
RLIM_SAVED_CUR	RLIM64_SAVED_CUR

<sys/stat.h>

struct stat	struct stat64
ino_t st_ino;	ino64_t st_ino;
off_t st_size;	off64_t st_size;
blkcnt_t st_blocks;	blkcnt64_t st_blocks;

<sys/statvfs.h>

struct statvfs	struct statvfs64
fsblkcnt_t f_blocks;	fsblkcnt64_t f_blocks;
fsblkcnt_t f_bfree;	fsblkcnt64_t f_bfree;
fsblkcnt_t f_bavial;	fsblkcnt64_t f_bavial;
fsfilcnt_t f_files;	fsfilcnt64_t f_files;
fsfilcnt_t f_ffree;	fsfilcnt64_t f_ffree;
fsfilcnt_t f_favail;	fsfilcnt64_t f_favail;

<sys/types.h>

off_t;	off64_t;
ino_t;	ino64_t;
blkcnt_t;	blkcnt64_t;
fsblkcnt_t;	fsblkcnt64_t;
fsfilcnt_t;	fsfilcnt64_t;

<unistd.h>

	_LFS64_LARGEFILE
	_LFS64_STDIO

<sys/unistd.h>

	_CS_LFS64_CFLAGS
	_CS_LFS64_LDFLAGS
	_CS_LFS64_LIBS
	_CS_LFS64_LINTFLAGS

System Interfaces

The following tables display the standard API and the corresponding transitional interfaces for 64-bit file offsets. The interfaces are grouped by header. The interface name and the affected data types are displayed in courier font.

<aio.h>

int aio_cancel(...,	int aio_cancel64(...,
struct aiocb *);	struct aiocb64 *);
int aio_error(	int aio_error64(
const struct aiocb *);	const struct aiocb64 *);
int aio_fsync(...,	int aio_fsync64(...,
struct aiocb *);	struct aiocb64 *);
int aio_read(struct aiocb *);	int aio_read64(struct aiocb64 *);
int aio_return(struct aiocb *);	int aio_return64(struct aiocb64 *);
int aio_suspend(	int aio_suspend64(

const struct aiocb *, ...);	const struct aiocb64 *, ...);
int aio_waitn(aiocb_t *[],	int aio_waitn64(aiocb64_t *[],
...);	...);
int aio_write(struct aiocb *);	int aio_write64(struct aiocb64 *);
int lio_listio(...,	int lio_listio64(...,
const struct aiocb *, ...);	const struct aiocb64 *, ...);

<dirent.h>

int alphasort(	int alphasort64(
const struct dirent **,	const struct dirent64 **,
const struct dirent **)	const struct dirent64 **)
struct dirent *readdir();	struct dirent64 *readdir64();
struct dirent *readdir_r();	struct dirent64 *readdir64_r();
int scandir(...,	int scandir64(...,
struct dirent *(*[]),	struct dirent64 *(*[]),
int (*)(const struct dirent *),	int (*)(const struct dirent64 *),
int (*)(const struct dirent **,	int (*)(const struct dirent64 **,
const struct dirent **))	const struct dirent64 **))

<fcntl.h>

int attropen();	int attropen64();
int creat();	int creat64();
int open();	int open64();
int openat();	int openat64();
int posix_fadvise()	int posix_fadvise64()
int posix_fallocate()	int posix_fallocate64()

<ftw.h>

int ftw(...,	int ftw64(...,
const struct stat *, ...);	const struct stat64 *, ...);
int nftw(..	int nftw64(...,
const struct stat *, ...);	const struct stat64 *, ...);

<libgen.h>

char *copylist(..., off_t);	char *copylist64(..., off64_t);

<stdio.h>

int fgetpos();	int fgetpos64();
FILE *fopen();	FILE *fopen64();
FILE *freopen();	FILE *freopen64();
int fseeko(..., off_t, ...);	int fseeko64(..., off64_t, ...);
int fsetpos(...,	int fsetpos64(...,
const fpos_t *);	const fpos64_t *);
off_t ftello();	off64_t ftello64();
FILE *tmpfile();	FILE *tmpfile64();

<stdlib.h>

int mkstemp();	int mkstemp64();

<sys/async.h>

int aioread(..., off_t, ...);	int aioread64(..., off64_t, ...);
int aiowrite(..., off_t, ...);	int aiowrite64(..., off64_t, ...);

<sys/dirent.h>

int getdents(..., dirent);	int getdents64(..., dirent64);

<sys/mman.h>

void mmap(..., off_t);	void mmap64(..., off64_t);

<sys/resource.h>

int getrlimit(...,	int getrlimit64(...,
struct rlimit *);	struct rlimit64 *);
int setrlimit(...,	int setrlimit64(...,
const struct rlimit *);	const struct rlimit64 *);

<sys/sendfile.h>

ssize_t sendfile(...,	ssize_t sendfile64(...,
off_t *, ...);	off64_t *, ...);
ssize_t sendfilev(..., const	ssize_t sendfilev64(..., const
struct sendfilevec *, ...);	struct sendfilevec64 *, ...);

<sys/stat.h>

int fstat(..., struct stat *);	int fstat64(..., struct stat64 *);
int fstatat(...,	int fstatat64(...,
struct stat *, int);	struct stat64 *, int);
int lstat(..., struct stat *);	int lstat64(..., struct stat64 *);
int stat(..., struct stat *);	int stat64(..., struct stat64 *);

<sys/statvfs.h>

int statvfs(...,	int statvfs64(...,
struct statvfs *);	struct statvfs64 *);
int fstatvfs(...,	int fstatvfs64(...,
struct statvfs *);	struct statvfs64 *);

<ucbinclude/stdio.h>

FILE *fopen()	FILE *fopen64()
FILE *freopen()	FILE *freopen64()

<ucbinclude/sys/dir.h>

int alphasort(	int alphasort64(
struct direct **,	struct direct64 **,
struct direct **);	struct direct64 **);
struct direct *readdir();	struct direct64 *readdir64();
int scandir(...,	int scandir64(...,
struct direct *(*[]);, ...);	struct direct64 *(*[]);, ...);

<unistd.h>

int lockf(..., off_t);	int lockf64(..., off64_t);
off_t lseek(..., off_t, ...);	off64_t lseek64(..., off64_t, ...);
int ftruncate(..., off_t);	int ftruncate64..., off64_t);
ssize_t pread(..., off_t);	ssize_t pread64..., off64_t);
ssize_t pwrite(..., off_t);	ssize_t pwrite64(..., off64_t);
int truncate(..., off_t);	int truncate64(..., off64_t);

関連項目

lfcompile(5), lfcompile64(5)

名前

lfcompile – large file compilation environment for 32-bit applications

機能説明

All 64-bit applications can manipulate large files by default. The methods described on this page allow 32-bit applications to manipulate large files.

In the large file compilation environment, source interfaces are bound to appropriate 64-bit functions, structures, and types. Compiling in this environment allows 32-bit applications to access files whose size is greater than or equal to 2 Gbyte (2^{31} bytes).

Each interface named *xxx*`()` that needs to access 64-bit entities to access large files maps to a *xxx*`64()` call in the resulting binary. All relevant data types are defined to be of correct size (for example, `off_t` has a typedef definition for a 64-bit entity).

An application compiled in this environment is able to use the *xxx*`()` source interfaces to access both large and small files, rather than having to explicitly utilize the transitional *xxx*`64()` interface calls to access large files. See the `lfcompile64(5)` manual page for information regarding the transitional compilation environment.

Applications can be compiled in the large file compilation environment by using the following methods:

- Use the `getconf(1)` utility with one or more of the arguments listed in the table below. This method is recommended for portable applications.

argument	purpose
`LFS_CFLAGS`	obtain compilation flags necessary to enable the large file compilation environment
`LFS_LDFLAGS`	obtain link editor options
`LFS_LIBS`	obtain link library names
`LFS_LINTFLAGS`	obtain lint options

- Set the compile-time flag `_FILE_OFFSET_BITS` to 64 before including any headers. Applications may combine objects produced in the large file compilation environment with objects produced in the transitional compilation environment, but must be careful with respect to interoperability between those objects. Applications should not declare global variables of types whose sizes change between compilation environments.

Access to Additional Large File Interfaces

The `fseek()` and `ftell()` functions *do not* map to functions named `fseek64()` and `ftell64()`; rather, the large file additions `fseeko()` and `ftello()`, have functionality identical to `fseek()` and `ftell()` and *do* map to the 64-bit functions `fseeko64()` and `ftello64()`. Applications wishing to access large files should use `fseeko()` and `ftello()` in place of `fseek()` and `ftell()`. See the `fseek(3C)` and `ftell(3C)` manual pages for information about `fseeko()` and `ftello()`.

Applications wishing to access `fseeko()` and `ftello()` as well as the POSIX and X/Open specification-conforming interfaces should define the macro `_LARGEFILE_SOURCE` to be 1 and set whichever feature test macros are appropriate to obtain the desired environment (see `standards(5)`).

使用例

In the following examples, the large file compilation environment is accessed by invoking the `getconf` utility with one of the arguments listed in the table above. The additional large file interfaces are accessed by specifying `–D_LARGEFILE_SOURCE`.

The examples that use the form of command substitution specifying the command within parentheses preceded by a dollar sign can be executed only in a POSIX-conforming shell such as the Korn Shell (see `ksh(1)`). In a shell that is not POSIX-conforming, such as the Bourne Shell (see `sh(1)`) and the C Shell (see `csh(1)`), the `getconf` calls must be enclosed within grave accent marks, as shown in the second example.

例 1 Compile a program with a “large” off_t that uses `fseeko()`, `ftello()`, and `yacc`.

The following example compiles a program with a “large” `off_t` and uses `fseeko()`, `ftello()`, and `yacc(1)`.

```
$ c89 -D_LARGEFILE_SOURCE                \
      -D_FILE_OFFSET_BITS=64 -o foo      \
      $(getconf LFS_CFLAGS) y.tab.c b.o  \
      $(getconf LFS_LDFLAGS)             \
      -ly $(getconf LFS_LIBS)
```

例 2 Compile a program with a “large” off_t that does not use `fseeko()` and `ftello()` and has no application specific libraries.

```
% c89 -D_FILE_OFFSET_BITS=64     \
     ` getconf LFS_CFLAG`S a.c   \
     ` getconf LFS_LDFLAG`S      \
     ` getconf LFS_LIB`S         \
```

例 3 Compile a program with a “default” off_t that uses `fseeko()` and `ftello()`.

```
$ c89 –D_LARGEFILE_SOURCE  a.c
```

関連項目

`csh(1)`, `getconf(1)`, `ksh(1)`, `yacc(1)`, `sh(1)`, `fseek(3C)`, `ftell(3C)`, `lf64(5)`, `lfcompile64(5)`, `standards(5)`

注意事項

Certain system-specific or non-portable interfaces are not usable in the large file compilation environment. Known cases are:

- Kernel data structures read from `/dev/kmem`.
- Interfaces in the kernel virtual memory library, `–lkvm`.
- Interfaces in the ELF access library, `–lelf`.
- Interfaces to `/proc` defined in `<procfs.h>`.
- The `ustat(2)` system call.

Programs that use these interfaces should not be compiled in the large file compilation environment. As a partial safeguard against making this mistake, including either of the <libelf.h> or <sys/procfs.h> header files will induce a compilation error when the large file compilation environment is enabled.

In general, caution should be exercised when using any separately-compiled library whose interfaces include data items of type `off_t` or the other redefined types either directly or indirectly, such as with '`struct stat`'. (The redefined types are `off_t`, `rlim_t`, `ino_t`, `blkcnt_t`, `fsblkcnt_t`, and `fsfilcnt_t`.) For the large file compilation environment to work correctly with such a library, the library interfaces must include the appropriate *xxx*`64()` binary entry points and must have them mapped to the corresponding primary functions when `_FILE_OFFSET_BITS` is set to 64.

Care should be exercised using any of the `printf()` or `scanf()` routines on variables of the types mentioned above. In the large file compilation environment, these variables should be printed or scanned using `long long` formats.

使用上の留意点

Symbolic formats analogous to those found in <sys/int_fmtio.h> do not exist for printing or scanning variables of the types that are redefined in the large file compilation environment.

名前

lfcompile64 – transitional compilation environment

機能説明

All 64-bit applications can manipulate large files by default. The transitional interfaces described on this page can be used by 32-bit and 64-bit applications to manipulate large files.

In the transitional compilation environment, explicit 64-bit functions, structures, and types are added to the API. Compiling in this environment allows both 32-bit and 64-bit applications to access files whose size is greater than or equal to 2 Gbyte (2^{31} bytes).

The transitional compilation environment exports all the explicit 64-bit functions (*xxx*`64()`) and types in addition to all the regular functions (*xxx*`()`) and types. Both *xxx*`()` and *xxx*`64()` functions are available to the program source. A 32-bit application must use the *xxx*`64()` functions in order to access large files. See the `lf64(5)` manual page for a complete listing of the 64-bit transitional interfaces.

The transitional compilation environment differs from the large file compilation environment, wherein the underlying interfaces are bound to 64-bit functions, structures, and types. An application compiled in the large file compilation environment is able to use the *xxx*`()` source interfaces to access both large and small files, rather than having to explicitly utilize the transitional *xxx*`64()` interface calls to access large files. See the `lfcompile(5)` manual page for more information regarding the large file compilation environment.

Applications may combine objects produced in the large file compilation environment with objects produced in the transitional compilation environment, but must be careful with respect to interoperability between those objects. Applications should not declare global variables of types whose sizes change between compilation environments.

For applications that do not wish to conform to the POSIX or X/Open specifications, the 64-bit transitional interfaces are available by default. No compile-time flags need to be set.

Access to Additional Large File Interfaces

Applications that wish to access the transitional interfaces as well as the POSIX or X/Open specification-conforming interfaces should use the following compilation methods and set whichever feature test macros are appropriate to obtain the desired environment (see `standards(5)`).

- Set the compile-time flag `_LARGEFILE64_SOURCE` to 1 before including any headers.
- Use the `getconf(1)` command with one or more of the following arguments:

argument	purpose
`LFS64_CFLAGS`	obtain compilation flags necessary to enable the transitional compilation environment
`LFS64_LDFLAGS`	obtain link editor options
`LFS64_LIBS`	obtain link library names
`LFS64_LINTFLAGS`	obtain lint options

使用例

In the following examples, the transitional compilation environment is accessed by invoking the `getconf` utility with one of the arguments listed in the table above. The additional large file interfaces are accessed either by specifying `–D_LARGEFILE64_SOURCE` or by invoking the `getconf` utility with the arguments listed above.

The example that uses the form of command substitution specifying the command within parentheses preceded by a dollar sign can be executed only in a POSIX-conforming shell such as the Korn Shell (see `ksh(1)`). In a shell that is not POSIX-conforming, such as the Bourne Shell (see `sh(1)`) and the C Shell (see `csh(1)`), the command must be enclosed within grave accent marks.

例 1 An example of compiling a program using transitional interfaces such as `lseek64()` and `fopen64()`:

```
$ c89 -D_LARGEFILE64_SOURCE        \
      $(getconf LFS64_CFLAGS) a.c  \
      $(getconf LFS64_LDFLAGS)     \
      $(getconf LFS64_LIBS)
```

例 2 An example of running lint on a program using transitional interfaces:

```
% lint -D_LARGEFILE64_SOURCE        \
     ` getconf LFS64_LINTFLAG`S ... \
     ` getconf LFS64_LIB`S
```

関連項目

`getconf(1)`, `lseek(2)`, `fopen(3C)`, `lf64(5)`, `standards(5)`

名前

locale – 言語や文化規約によるユーザー環境のサブセット

機能説明

locale とは、言語や文化規約によるユーザー環境のサブセットを定義したものです。1 つ以上のカテゴリで構成されます。各カテゴリはその名前によって識別され、システムコンポーネントの動作の特定の側面を制御します。カテゴリ名は次の環境変数名に対応しています:

LC_CTYPE　文字の分類および大文字/小文字の変換。

LC_COLLATE　照合順序。

LC_TIME　日付と時間の書式。

LC_NUMERIC　数値の書式設定。

LC_MONETARY　通貨の書式設定。

LC_MESSAGES　情報メッセージや診断メッセージおよび対話式応答の書式。

これらの標準ユーティリティーは、各ユーティリティーの「環境変数」セクションに定義されているように、現在のロケールに基づいて動作します。一部の C 言語関数の動作も、setlocale(3C) の最後の呼び出しで定義されているように、現在のロケールに基づいて変更されます。

実装によって提供される以外のロケールは、localedef(1) ユーティリティーを通じてアプリケーションで作成できます。環境変数の使用時にロケールの指定に使われる値は、そのロケールが作成されたときに localedef の *name* オペランドとして指定された文字列になります。文字列「C」および「POSIX」は POSIX ロケールの識別子として予約されています。

必要なロケールをアプリケーションで選択するには、適切な値を指定して setlocale() 関数を呼び出します。この関数を次のような空の文字列で呼び出した場合:

```
setlocale(LC_ALL, "");
```

対応する環境変数の値が使用されます。環境変数が未設定か、または空の文字列に設定されている場合は、setlocale() 関数によって適切な環境が設定されます。

ロケール定義

ロケールは、localedef ユーティリティーで受け入れられるファイル形式を使って記述できます。

ロケール定義ファイルにはロケールカテゴリのソース定義が 1 つ以上含まれている必要があり、同じロケールカテゴリの定義が複数あってはいけません。

カテゴリソース定義は、カテゴリヘッダー、カテゴリ本体、およびカテゴリトレーラで構成されます。カテゴリヘッダーは、文字 LC_ で始まる、カテゴリ名を表

す文字列で構成されます。カテゴリトレーラは、文字列 END とそれに続く 1 つ以上の空白文字、および対応するカテゴリヘッダーで使用される文字列で構成されます。

カテゴリ本体は、1 行以上のテキストで構成されます。各行には識別子が 1 つ含まれ、そのあとに必要に応じて 1 つ以上のオペランドが続きます。識別子は、特定のロケール要素を識別するキーワードか、または照合要素のどちらかです。ロケール内の各キーワードには一意の名前を付ける必要があります (つまり、2 つのカテゴリに一般的な名前のキーワードを含めることはできません)。キーワードを文字 LC_ で始めることはできません。識別子とオペランドは 1 つ以上の空白文字で区切る必要があります。

オペランドは、文字、照合要素、または文字列である必要があります。文字列は二重引用符 (“) で囲む必要があります。文字列内のリテラルの二重引用符の前には、次のように <*escape character*> を付ける必要があります。キーワードのあとに複数のオペランドが続く場合は、それらのオペランドをセミコロン (;) で区切る必要があります。セミコロンの前後に空白文字を入れてもかまいません。

ファイル内の最初のカテゴリヘッダーの前に、コメント文字を変更する行を付けることができます。それは 1 列目から始まり、次のような形式になります:

"comment_char %c\n",<*comment character*>

コメント文字はデフォルトで番号記号 (#) になっています。先頭位置にある空白行と、<*comment character*> を含む行は無視されます。

ファイル内の最初のカテゴリヘッダーの前に、そのファイル内で使用されるエスケープ文字を変更する行を付けることができます。それは 1 列目から始まり、次のような形式になります:

"escape_char %c\n",<*escape character*>

エスケープ文字はデフォルトでバックスラッシュになっています。

行を継続するには、その行の最後の文字としてエスケープ文字を配置します。この継続文字は入力から破棄されます。実装では、長さが {LINE_MAX} バイトを超える継続行のいずれの部分も受け入れる必要はありませんが、継続行の累積の長さには制限を設けていません。コメント行は、エスケープされた改行文字を使って次の行に継続することはできません。

個々の文字、文字列内の文字、および照合要素は、次に定義しているように、シンボリック名を使用して表現する必要があります。また、文字は文字そのものを使用して表現することも、8 進定数、16 進定数、または 10 進定数として表現することもできます。シンボリック以外の表記法を使用すると、結果として得られるロケール定義は多くの場合、システム間で移植できなくなります。左山括弧 (<) はシンボ

リック名の開始を意味する予約シンボルであり、それ自体を表現するために使用するときは、その前にエスケープ文字を付ける必要があります。文字表現には次の規則が適用されます:

1. 山括弧 < と > で囲んだシンボリック名で文字を表現できます。シンボリック名 (山括弧を含む) は、localedef -f オプションで指定された charmap ファイルに定義されているシンボリック名と正確に一致する必要があり、charmap ファイル内のシンボリック名に関連付けられた値から特定される文字値に置き換えられます。charmap ファイルに存在しないシンボリック名を使用すると、エラーが発生しますが、カテゴリが LC_CTYPE または LC_COLLATE の場合は例外で、その場合は警告状態になります (エラーや警告がもたらす影響については localedef(1) を参照)。collating-element または collating-symbol セクションでのシンボリック名の指定のうち、charmap ファイル (存在する場合) 内でシンボリック名が重複しているものはエラーです。シンボリック名の内部でのエスケープ文字または右山括弧の使用は、文字の前にエスケープ文字を付ける場合を除いて無効です。

   例:

   ```
   <C>;<c-cedilla> "<M><a><y>"
   ```

2. 文字はその文字自体で表現することができ、その場合、文字の値は実装に依存します。文字列の内部では、二重引用符文字、エスケープ文字、および右山括弧文字についてはエスケープして (エスケープ文字を前に付ける)、文字自体として解釈されるようにする必要があります。文字列の外側では、文字

   ,　　;　　<　　> *escape_char*

   をエスケープして文字自体として解釈されるようにする必要があります。

   例:

   ```
   c       "May"
   ```

3. 文字を 8 進定数で表現できます。8 進定数は、エスケープ文字のあとに 8 進数字が 2 つ以上続く形式で指定されます。各定数はバイト値を表します。複数バイト値は、定数の最後の文字が最下位バイトを示すようなバイト順序で指定された連結定数によって表現できます。

   例:

   ```
   \143;\347;\143\150    "\115\141\171"
   ```

4. 文字を 16 進定数で表現できます。16 進定数は、エスケープ文字のあとに x が 1 つ続き、そのあとに 16 進数字が 2 つ以上続く形式で指定されます。各定数はバイト値を表します。複数バイト値は、定数の最後の文字が最下位バイトを示すようなバイト順序で指定された連結定数によって表現できます。

   例:

   ```
   \x63;\xe7;\x63\x68    "\x4d\x61\x79"
   ```

5. 文字を 10 進定数で表現できます。10 進定数は、エスケープ文字のあとに d が 1 つ続き、そのあとに 10 進数字が 2 つ以上続く形式で指定されます。各定数はバイト値を表します。複数バイト値は、定数の最後の文字が最下位バイトを示すようなバイト順序で指定された連結定数によって表現できます。

   例:

   ```
   \d99;\d231;\d99\d104   "\d77\d97\d121"
   ```

   シンボリック名、文字自体、8 進定数、10 進定数、または 16 進定数のどれを使用しても、ロケール定義が作成される文字セット内に存在する文字しか指定できません。charmap ファイルが存在する場合は、その charmap 内に定義されている文字のみを 8 進定数、10 進定数、または 16 進定数を使って指定できます。charmap ファイルに存在しないシンボリック名は、上記の項目 1 で規定されているように、指定することはできますが、無視されます。

LC_CTYPE

LC_CTYPE カテゴリは、文字の分類、大文字/小文字の変換、およびその他の文字属性を定義します。また、一連の文字を、省略記号 (...) を表す 3 つの隣接するピリオドで表現できます。省略記号の指定は、省略記号の前と後にある値の間にあるすべての値が有効な文字であるという意味に解釈されます。省略記号の指定はエンコードされた単一の文字セット内、つまり同じサイズの文字のグループ内でのみ有効です。省略記号は、その省略記号の前にある文字のエンコードされた値よりも大きく、その省略記号の後ろにある文字のエンコードされた値よりも小さいエンコードされた値を持つすべての文字をリストに含めるものと解釈されます。

例:

```
\x30;...;\x39;
```

エンドポイント間にあるエンコードされた値を持つすべての文字を文字クラスに含めます。

次のキーワードが認識されています。説明の中で、「自動的に含まれる」という用語は参照文字のいずれかを含めても、省略してもエラーにならないことを意味します。

文字クラス digit、xdigit、lower、upper、および space には、自動的に含まれる文字セットがあります。これらは、文字値 (つまりエンコーディング) が実装のデフォルト値と異なっている場合にのみ指定する必要があります。

upper

大文字に分類される文字を定義します。

POSIX ロケールでは、26 個の大文字が含まれます:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

ロケール定義ファイルでは、キーワード cntrl、digit、punct、または space で指定された文字は指定できません。大文字 A-Z がこのクラスに自動的に含まれます。

`lower` 小文字に分類される文字を定義します。POSIX ロケールでは、26 個の小文字が含まれます:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

ロケール定義ファイルでは、キーワード `cntrl`、`digit`、`punct`、または `space` で指定された文字は指定できません。移植可能な文字セットの小文字 `a - z` がこのクラスに自動的に含まれます。

`alpha` アルファベット文字に分類される文字を定義します。

POSIX ロケールでは、クラス `upper` と `lower` に含まれるすべての文字が含まれます。

ロケール定義ファイルでは、キーワード `cntrl`、`digit`、`punct`、または `space` で指定された文字は指定できません。`upper` または `lower` のどちらかに分類された文字がこのクラスに自動的に含まれます。

`digit` 数字に分類される文字を定義します。

POSIX ロケールでは、

```
0 1 2 3 4 5 6 7 8 9
```

のみが含まれます。

ロケール定義ファイルでは、数字 `0`、`1`、`2`、`3`、`4`、`5`、`6`、`7`、`8`、および `9` のみを、昇順の連続する数値で指定できます。移植可能な文字セットの数字 `0 - 9` がこのクラスに自動的に含まれます。

文字クラス `digit` の定義に必要なのはそれらの 10 文字のみです。数字を定義するそれらの文字は指定できますが、代替の数字 (ヒンディー語や漢字など) をここで指定することはできません。

`alnum` 文字および数字に分類される文字を定義します。`alpha` および `digit` キーワードで指定された文字のみが指定されます。キーワード `alpha` および `digit` で指定された文字がこのクラスに自動的に含まれます。

`space` 空白文字に分類される文字を定義します。

POSIX ロケールでは、少なくとも文字 `SPACE`、`FORMFEED`、`NEWLINE`、`CARRIAGE RETURN`、`TAB`、および `VERTICAL TAB` が含まれます。

ロケール定義ファイルでは、キーワード upper、lower、alpha、digit、graph、または xdigit で指定された文字は指定できません。移植可能な文字セットの文字 SPACE、FORMFEED、NEWLINE、CARRIAGE RETURN、TAB、および VERTICAL TAB と、クラス blank に含まれるすべての文字がこのクラスに自動的に含まれます。

`cntrl` 制御文字に分類される文字を定義します。

POSIX ロケールでは、クラス alpha または print 内の文字は含まれません。

ロケール定義ファイルでは、キーワード upper、lower、alpha、digit、punct、graph、print、または xdigit で指定された文字は指定できません。

`punct` 句読文字に分類される文字を定義します。

POSIX ロケールでは、空白文字も、クラス alpha、digit、または cntrl 内のどの文字も含まれません。

ロケール定義ファイルでは、キーワード upper、lower、alpha、digit、cntrl、xdigit で指定された文字と空白文字は指定できません。

`graph` 出力可能文字に分類される文字 (空白文字は含まない) を定義します。

POSIX ロケールでは、クラス alpha、digit、および punct 内の文字はすべて含まれますが、クラス cntrl 内の文字は含まれません。

ロケール定義ファイルでは、キーワード upper、lower、alpha、digit、xdigit、および punct で指定された文字がこのクラスに自動的に含まれます。キーワード cntrl で指定された文字は指定できません。

`print` 出力可能文字に分類される文字 (空白文字を含む) を定義します。

POSIX ロケールでは、クラス graph 内の文字はすべて含まれますが、クラス cntrl 内の文字は含まれません。

ロケール定義ファイルでは、キーワード upper、lower、alpha、digit、xdigit、および punct で指定された文字と空白文字がこのクラスに自動的に含まれます。キーワード cntrl で指定された文字は指定できません。

`xdigit` 16 進数字に分類される文字を定義します。

POSIX ロケールでは:

```
0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f
```

のみが含まれます。

ロケール定義ファイルでは、クラス `digit` で指定された文字のみを昇順の連続する数値で指定し、そのあとに 16 進数字 10 - 15 (15 を含む) を表す 6 文字のセット (各セットは昇順に指定) を 1 つ以上続けることができます (`A`、`B`、`C`、`D`、`E`、`F`、`a`、`b`、`c`、`d`、`e`、`f` など)。移植可能な文字セットの数字 `0 - 9`、大文字 `A - F`、および小文字 `a - f` がこのクラスに自動的に含まれます。

文字クラス `xdigit` の定義では、文字クラス `digit` に含まれる文字もここに含まれることが必要です。

`blank` 空白文字に分類される文字を定義します。

POSIX ロケールでは、空白文字とタブ文字のみが含まれます。

ロケール定義ファイルでは、空白文字とタブ文字がこのクラスに自動的に含まれます。

`charclass` 1 つ以上のロケール固有の文字クラス名をセミコロンで区切られた文字列として定義します。その後、それぞれの名前付き文字クラスを `LC_CTYPE` 定義内に定義できます。文字クラス名は、移植可能なファイル名文字セットからの 1 バイト以上 `{CHARCLASS_NAME_MAX}` バイト以下の英数字で構成されます。文字クラス名の最初の文字を数字にすることはできません。その名前をこのドキュメントに定義されている `LC_CTYPE` キーワードのいずれとも同じにすることはできません。

`charclass-name` 名前付きのロケール固有の文字クラスに属するものとして分類される文字を定義します。POSIX ロケールでは、ロケール固有の名前付き文字クラスは必要ありません。クラス名が `charclass` キーワードで定義されているが、その後に文字が割り当てられていない場合、これはエラーではなく、どの文字も属していないクラスを表しています。`charclass-name` は、`wctype(3C)` 関数の *property* 引数として、正規表現とシェルパターンマッチングのブラケット表現で、および `tr(1)` コマンドによって使用できます。

`toupper` 小文字から大文字へのマッピングを定義します。

POSIX ロケールでは、少なくとも 26 個の小文字:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

が対応する 26 個の大文字にマップされます:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

ロケール定義ファイルでは、このオペランドはセミコロンで区切られた文字ペアで構成されます。各文字ペアの文字はコンマで区切られ、ペアは括弧で囲まれます。各ペアの最初の文字は小文字で、2 番目の文字は対応する大文字です。キーワード `lower` および `upper` で指定された文字のみを指定できます。移植可能な文字セットの小文字 `a - z` とそれに対応する大文字 `A - Z` がこのマッピングに自動的に含まれますが、`toupper` キーワードがロケール定義で省略されている場合だけは除きます。

`tolower` 大文字から小文字へのマッピングを定義します。

POSIX ロケールでは、少なくとも 26 個の大文字:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

が対応する 26 個の小文字にマップされます:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

ロケール定義ファイルでは、このオペランドはセミコロンで区切られた文字ペアで構成されます。各文字ペアの文字はコンマで区切られ、ペアは括弧で囲まれます。各ペアの最初の文字は大文字で、2 番目の文字は対応する小文字です。キーワード `lower` および `upper` で指定された文字のみを指定できます。`tolower` キーワードがロケール定義で省略されている場合、マッピングは `toupper` に指定されたものの逆マッピングになります。

LC_COLLATE

LC_COLLATE カテゴリは、多数のユーティリティー (`sort(1)`、`uniq(1)` など)、正規表現照合 (`regex(5)` を参照)、そして `strcoll(3C)`、`strxfrm(3C)`、`wcscoll(3C)`、および `wcsxfrm(3C)` 関数のための照合順序定義を提供します。

照合順序定義では、ロケール内の照合要素 (文字および複数文字の照合要素) 間の相対的な順序を定義します。この順序は、照合値によって、つまり各要素に 1 つ以上の照合値 (照合ウェイトとも呼ばれる) を割り当てることによって表されます。次の機能があります:

1. 複数文字の照合要素。複数文字の照合要素の指定 (つまり、1 つのエンティティーとして照合される 2 つ以上の文字のシーケンス)。
2. ユーザー定義の照合要素順序。各照合要素には、文字 (基本) 照合順序におけるその順番を定義する照合値が割り当てられます。この順序付けは、正規表現やパターンマッチングによって使用され、また照合ウェイトが明示的に指定されていない場合はソートで使用される照合ウェイトとしても使用されます。

3. 複数のウェイトと等価クラス。照合要素には、ソートで使用する照合ウェイトを 1 つ以上 (制限値 {COLL_WEIGHTS_MAX} まで) 割り当てることができます。以降は、最初のウェイトをプライマリウェイトと呼びます。
4. 1 対多のマッピング。1 つの文字が一連の照合要素にマッピングされます。
5. 等価クラス定義。2 つ以上の照合要素が同じ照合値 (プライマリウェイト) を持ちます。
6. ウェイトによる順序付け。2 つの文字列を比較してその相対的な順序を判定する場合、それらの 2 つの文字列は最初に一連の照合要素に分割されます。次に、それらの要素の相対的なプライマリウェイトに従って、連続する要素ペア内の要素が比較されます。等しい場合、複数のウェイトが割り当てられていれば、照合要素ペアの 1 つが等しくないと判断されるか、またはウェイトが使い果たされるまで、相対的な以降のウェイトに従って照合要素ペアが再度比較されます。

照合順序定義では、次のキーワードが認識されています。それらについては、以降のセクションで詳しく説明します。

`copy`	このカテゴリの定義として使用される、既存のロケールの名前を指定します。このキーワードを指定した場合は、ほかのキーワードは指定しません。
`collating-element`	複数文字の照合要素を表す照合要素シンボルを定義します。このキーワードは省略可能です。
`collating-symbol`	照合順序文で使用する照合シンボルを定義します。このキーワードは省略可能です。
`order_start`	照合規則を定義します。この文のあとに、文字の照合値と照合ウェイトを照合要素に割り当てる、1 つ以上の照合順序文が続きます。
`order_end`	照合順序文の終了を指定します。

collating-element キーワード

文字セット内の照合要素に加えて、`collating-element` キーワードを使用すると、複数文字の照合要素を定義できます。構文は次のとおりです:

```
"collating-element %s from \"%s\"\n",<collating-symbol>,<string>
```

<*collating-symbol*> オペランドは山括弧 (< と >) で囲まれたシンボリック名であり、現在の charmap ファイル (存在する場合) 内のシンボリック名、またはこの照合定義内で定義されているほかのシンボリック名と重複してはいけません。string オペランドは、1 つのエンティティーとして照合される 2 文字以上の文字列です。このキーワードで定義される <*collating-element*> は、`LC_COLLATE` カテゴリでのみ認識されます。

例:

collating-element <ch> from "<c><h>"
collating-element <e-acute> from "<acute><e>"
collating-element <ll> from "ll"

collating-symbol キーワード

このキーワードは、照合順序文、つまり order_start キーワードと order_end キーワードの間で使用するシンボルの定義に使われます。構文は次のとおりです:

"collating-symbol %s\n",<*collating-symbol*>

<*collating-symbol*> は山括弧 (< と >) で囲まれたシンボリック名であり、現在の charmap ファイル (存在する場合) 内のシンボリック名、またはこの照合定義内で定義されているほかのシンボリック名と重複してはいけません。

このキーワードによって定義される collating-symbol は、LC_COLLATE カテゴリでのみ認識されます。

例:

collating-symbol <UPPER_CASE>
collating-symbol <HIGH>

collating-symbol キーワードは、文字順序シーケンス内の相対位置に関連付けることができるシンボリック名を定義します。そのようなシンボリック名はいずれの照合要素も表しませんが、ウェイトとして使用できます。

order_start キーワード

order_start キーワードは、照合順序エントリの前に置く必要があり、この照合順序定義のウェイトの数とその他の照合規則を定義します。

order_start キーワードの構文は次のとおりです:

"order_start %s;%s;...;%s\n",<*sort-rules*>,<*sort-rules*>

order_start キーワードのオペランドはオプションです。存在する場合、オペランドは文字列が比較されるときに適用される規則を定義します。オペランドの数は、各要素に割り当てられているウェイトの数を定義します。オペランドが存在しない場合は、1 つの forward オペランドが想定されます。存在する場合は、最初のオペランドが最初の (プライマリ) ウェイトを使って文字列を比較するときに適用される規則を定義し、2 番目のオペランドが 2 番目のウェイトを使って文字列を比較するときに適用される規則を定義し、以下同様になります。オペランドはセミコロン (;) で区切ります。各オペランドは、コンマ (,) で区切った 1 つ以上の照合指令で構成されます。オペランドの数が制限値 {COLL_WEIGHTS_MAX} を超えると、ユーティリティーによって警告メッセージが発行されます。次の指令がサポートされています:

forward　ウェイトレベルの比較演算が文字列の先頭から文字列の終わりに向かって行われることを指定します。

`backward` ウェイトレベルの比較演算が文字列の終わりから文字列の先頭に向かって行われることを指定します。

`position` ウェイトレベルの比較演算で、`IGNORE` の対象とならない文字列内の要素の相対位置が考慮されることを指定します。比較の開始から `IGNORE` の対象となる最少の照合要素の後ろにある、`IGNORE` の対象とならない要素を含む文字列が最初に照合されます。両方の文字列内で、`IGNORE` の対象とならない文字が同じ相対位置に含まれている場合、それらの要素に割り当てられた照合値によって順序付けが決められます。同等の場合、`IGNORE` の対象とならない後続の文字が同様に考慮されます。

指令 `forward` および `backward` は相互に排他的です。

例:

```
order_start     forward;backward
```

オペランドを指定しない場合は、1 つの `forward` オペランドが想定されます。

照合順序

`order_start` キーワードのあとには照合識別子エントリが続きます。照合要素エントリの構文は次のとおりです:

"`%s %s;%s;...;%s\n`"<*collating-identifier*>,<*weight*>,<*weight*>,...

各 *collating-identifier* は、上記の「ロケール定義」で説明している文字、<*collating-element*>、<*collating-symbol*>、省略記号、または特殊シンボル `UNDEFINED` で構成されます。照合要素が指定される順序によって、各照合要素がそれに続く要素に比べて小さくなるように文字順序シーケンスが決められます。NUL 文字はほかのどの文字と比べても小さくなります。

<*collating-element*> は、複数文字の照合要素を指定するために使用され、<*collating-element*> で指定された文字シーケンスが 1 つの単位として、その場所によって指定された相対的な順序で照合されることを示します。

<*collating-symbol*> は、ウェイトで使用する、相対的な順序での位置を定義するために使用します。 <*collating-symbol*> ではウェイトは指定されません。

省略記号は、一連の文字がそれらのエンコードされた文字値に従って照合されることを指定します。それは、現在のコード化文字セット内にある、前の行の文字の値よりも大きく、次の行の文字の値よりも小さいコード化文字セット値を持つすべての文字が、それらのコード化文字セット値に従って昇順に、前の文字と次の文字間に文字照合順序で配置されることを示すものと解釈されます。現在のコード化文字セット内で、最初の省略記号は前の行に NUL 文字が指定されているかのように解釈され、末尾の省略記号は次の行にもっとも大きいコード化文字セット値が指定されているかのように解釈されます。現在のコード化文字セット内で前の行と次の行

に文字が指定されていない場合、省略記号は無効とみなされます。省略記号の使用により、定義が特定のコード化文字セットに結び付けられ、実装間で定義を移植できないようにすることができます。

シンボル UNDEFINED は、明示的にも、または省略記号によっても指定されていないすべてのコード化文字セット値を含むものと解釈されます。このような文字は、文字照合順序内の、このシンボルで指示された位置に、それらのコード化文字セット値に従って昇順に挿入されます。UNDEFINED シンボルが指定されず、現在のコード化文字セットに、このセクションで定められていない文字が含まれている場合、ユーティリティーによって警告メッセージが発行され、そのような文字は文字照合順序の最後に置かれます。

各照合要素のオプションのオペランドは、照合要素のプライマリ、セカンダリ、またはそれ以降のウェイトを定義するために使用されます。最初のオペランドは相対的なプライマリウェイトを指定し、2 番目は相対的なセカンダリウェイトを指定し、以下同様になります。2 つ以上の照合要素に同じウェイトを割り当てることができます。つまり、それらが同じプライマリウェイトを持つ場合、それらは同じ等価クラスに属します。order_start キーワードで対応するレベルに対して position 照合指令が指定されていなければ、照合はウェイトレベルごとに、IGNORE の対象となる要素が削除されたかのように動作します。そして、次の要素ペアがそれらの要素の相対的なウェイトに従って比較されます。2 つの文字列を比較して等しい場合は、制限値 {COLL_WEIGHTS_MAX} に達するまで、次のウェイトレベルに対してこのプロセスが繰り返されます。

ウェイトは、上記の「ロケール定義」で説明している文字、<*collating-symbol*>、<*collating-element*>、省略記号、または特殊シンボル IGNORE として表されます。1 つの文字、< *collating–symbol*>、または <*collating–element*> は、文字または文字自体ではなく、文字またはシンボルの文字照合順序における相対位置を表します。このように、絶対値をウェイトに割り当てるのではなく、特定のウェイトが、文字照合順序での順序に基づいて照合要素に割り当てられた相対的な順序値を使って表されます。

1 対多のマッピングは、2 つ以上の連結された文字またはシンボリック名を指定することで示されます。たとえば、文字 <eszet> に文字列 “<s><s>” をウェイトとして指定した場合は、文字 <eszet> のすべてのオカレンスが <s><s> によって置き換えられたかのように比較が行われます (<s> には照合ウェイト <s> が割り当てられているものとします)。<eszet> と <s><s> を等価クラスとして定義する必要がある場合は、文字列 ss に対して照合要素を定義する必要があります。

省略記号で指定されたすべての文字にはデフォルトで、文字の相対的な順序に等しい、一意のウェイトが割り当てられます。明示的または暗黙的な UNDEFINED 特殊シンボルで指定された文字には、デフォルトで同じプライマリウェイトが割り当てられます (つまり、それらは同じ等価クラスに属します)。ウェイトとしての省略記号は、シーケンス内の各文字が、文字照合順序でのそれらの文字の相対的な順序に等

しい、一意のウェイトを持つことを意味するものと解釈されます。省略記号のウェイトとしての使用は、照合要素が省略記号でも特殊シンボル UNDEFINED でもない場合、エラーとして扱われます。

ウェイトとしての特殊なキーワード IGNORE は、IGNORE が指定されているレベルのウェイトを使用して文字列を比較するときに、その照合要素が無視される、つまり文字列にその照合要素が含まれていなかったかのようになることを示します。正規表現およびパターンマッチングでは、プライマリウェイトで IGNORE の対象となるすべての文字によって等価クラスが形成されます。

空のオペランドは、照合要素そのものとして解釈されます。

たとえば、順序文:

```
<a>     <a>;<a>
```

は次と同じです:

```
<a>
```

省略記号は、照合要素が省略記号だった場合はオペランドとして使用でき、その省略記号によって定義された各文字の値として解釈されます。

このセクションで定義されている照合順序は、正規表現におけるブラケット表現の解釈を定義します。

例:

```
order_start                             forward;backward
UNDEFINED                               IGNORE;IGNORE
<LOW>
<space>                                 <LOW>;<space>
. . .                                   <LOW>;. . .
<a>                                     <a>;<a>
<a-acute>                               <a>;<a-acute>
<a-grave>                               <a>;<a-grave>
<A>                                     <a>;<A>
<A-acute>                               <a>;<A-acute>
<A-grave>                               <a>;<A-grave>
<ch>                                    <ch>;<ch>
```

```
<Ch>                                    <ch>;<Ch>

<s>                                     <s>;<s>

<eszet>                                 "<s><s>";"<eszet><eszet>"

order_end
```

この例は、次のように解釈されます:

1. UNDEFINED は、この定義内に (明示的に、または省略記号で) 指定されていないすべての文字が照合の際に無視されることを意味します。正規表現の際には、それらは先頭に順序付けられます。
2. <space> と <a> の間にあるすべての文字は同じプライマリ等価クラスと、エンコードされた序数値に基づく個別のセカンダリウェイトを持ちます。
3. 大文字または小文字の a に基づいた文字はすべて同じ 1 次等価クラスに属します。
4. 複数文字の照合要素 <ch> は照合シンボル <ch> で表され、複数文字の照合要素 <Ch> と同じ 1 次等価クラスに属します。

order_end キーワード

照合順序エントリは、order_end キーワードで終了させる必要があります。

LC_MONETARY

LC_MONETARY カテゴリは、通貨の数値情報の書式設定に使用される規則と記号を定義します。この情報は、localeconv(3C) 関数によって入手できます

ロケールのこのカテゴリには、次の項目が定義されています。項目名は、ロケールの定義時に localedef(1) ユーティリティーによって認識されるキーワードです。それらは、<locale.h> に定義されている lconv 構造のメンバー名にも似ています。localeconv 関数は、未定義の整数項目に対して {CHAR_MAX} を返し、未定義またはサイズが 0 の文字列項目に対して空の文字列 ("") を返します。

ロケール定義ファイルでは、オペランドは文字列です。一部のキーワードでは、文字列に整数しか含めることができません。提供されていないキーワード、空の文字列 ("") に設定された文字列値、–1 に設定された整数のキーワードは、その値がロケールで使用できないことを示すために使われます。

int_curr_symbol	国際通貨記号。オペランドは 4 文字からなる文字列であり、最初の 3 文字には ISO 4217 標準に指定されたものに合致した英字の国際通貨記号が含まれます。4 番目の文字は、国際通貨記号と金額を区切るために使われる文字です。
currency_symbol	ローカルの通貨記号として使用される文字列。
mon_decimal_point	オペランドは、金額を書式設定する際に小数点 (基数文字) として使用される記号を含む文字列です。

`mon_thousands_sep` オペランドは、書式設定された金額の中で小数点の左側にある数字の位取り記号として使用される記号を含む文字列です。

`mon_grouping` 書式設定された金額の中で数字の位取りの桁数を定義します。オペランドは、セミコロンで区切られた一連の整数です。各整数は位取りの桁数を指定します。最初の整数は小数点のすぐ左側の桁数を定義し、以降の整数はその左側の桁数を定義します。最後の整数が –1 でない場合、その前の桁数 (存在する場合) が残りの数字に対して繰り返し使用されます。最後の整数が –1 の場合、これ以上の位取りは行われません。

`mon_grouping` キーワードの解釈の例を次に示します。書式設定される値は 123456789 で、`mon_thousands_sep` が ’ であると仮定して、次の表にその結果が表示されています。3 列目には、この位取りに対応するために `localeconv` 関数で使用される、ISO C 標準に含まれているのと同等の文字列が表示されています。

`mon_grouping`	Formatted Value	ISO C String
3;-1	123456’789	"\3\177"
3	123’456’789	"\3"
3;2;-1	1234’56’789	"\3\2\177"
3;2	12’34’56’789	"\3\2"
-1	1234567898	"\177"

これらの例では、`{CHAR_MAX}` の 8 進値は 177 です。

`positive_sign` 書式設定された負でない金額を示すために使用される文字列。

`negative_sign` 書式設定された負の金額を示すために使用される文字列。

`int_frac_digits` `int_curr_symbol` を使用して書式設定された金額の中で記述される小数点以下の桁数 (小数点の右側の桁数) を表す整数。

`frac_digits` `currency_symbol` を使用して書式設定された金額の中で記述される小数点以下の桁数 (小数点の右側の桁数) を表す整数。

`p_cs_precedes` SUSv3 標準に準拠したアプリケーションでは、`currency_symbol` が、負でない金額値の前に置かれる場合は `1` に設定され、その記号がその値のあとに続く場合は `0` に設定される整数。

SUSv3 標準に準拠しないアプリケーションでは、currency_symbol または int_currency_symbol が、負でない金額値の前に置かれる場合は 1 に設定され、その記号がその値のあとに続く場合は 0 に設定される整数。

p_sep_by_space

SUSv3 標準に準拠したアプリケーションでは、currency_symbol と負でない金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

SUSv3 標準に準拠しないアプリケーションでは、currency_symbol または int_curr_symbol と負でない金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

n_cs_precedes

SUSv3 標準に準拠したアプリケーションでは、currency_symbol が、負の金額値の前に置かれる場合は 1 に設定され、その記号がその値のあとに続く場合は 0 に設定される整数。

SUSv3 標準に準拠しないアプリケーションでは、currency_symbol または int_currency_symbol が、負の金額値の前に置かれる場合は 1 に設定され、その記号がその値のあとに続く場合は 0 に設定される整数。

n_sep_by_space

SUSv3 標準に準拠したアプリケーションでは、currency_symbol と負の金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

SUSv3 標準に準拠しないアプリケーションでは、currency_symbol または int_curr_symbol と負の金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

p_sign_posn

負でない金額値における、positive_sign の位置を示す値に設定される整数。p_sign_posn と n_sign_posn の両方に対して、次の整数値が認識されています:

SUSv3 標準に準拠したアプリケーションでは:

0 金額と currency_symbol を括弧で囲みます。

1 金額と currency_symbol の前に符号文字列が置かれます。

2 金額と currency_symbol のあとに符号文字列が続きます。

3 currency_symbol の前に符号文字列が置かれます。

4 currency_symbol のあとに符号文字列が続きます。

SUSv3 標準に準拠しないアプリケーションでは:

0 金額と currency_symbol または int_curr_symbol を括弧で囲みます。

1 金額と currency_symbol または int_curr_symbol の前に符号文字列が置かれます。

2 金額と currency_symbol または int_curr_symbol のあとに符号文字列が続きます。

3 currency_symbol または int_curr_symbol の前に符号文字列が置かれます。

4 currency_symbol または int_curr_symbol のあとに符号文字列が続きます。

n_sign_posn 書式設定された負の金額値における、negative_sign の位置を示す値に設定される整数。

int_p_cs_precedes int_curr_symbol が、負でない金額値の前に置かれる場合は 1 に設定され、その記号がその値のあとに続く場合は 0 に設定される整数。

int_n_cs_precedes int_curr_symbol が、負の金額値の前に置かれる場合は 1 に設定され、その記号がその値のあとに続く場合は 0 に設定される整数。

int_p_sep_by_space int_curr_symbol と負でない金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

int_n_sep_by_space int_curr_symbol と負の金額値が空白で区切られない場合は 0 に設定され、その記号とその値が空白で区切られる場合は 1 に設定され、その記号と符号文字列が隣接する場合に両者が空白で区切られる場合は 2 に設定される整数。

`int_p_sign_posn` 国際書式で書式設定された正の金額値における、`positive_sign` の位置を示す値に設定される整数。`int_p_sign_posn` と `int_n_sign_posn` に対して、次の整数値が認識されています:

- `0` 金額と `int_curr_symbol` を括弧で囲みます。
- `1` 金額と `int_curr_symbol` の前に符号文字列が置かれます。
- `2` 金額と `int_curr_symbol` の前に符号文字列が置かれます。
- `3` `int_curr_symbol` の前に符号文字列が置かれます。
- `4` `int_curr_symbol` のあとに符号文字列が続きます。

`int_n_sign_posn` 国際書式で書式設定された負の金額値における、`negative_sign` の位置を示す値に設定される整数。

さまざまな組み合わせの結果を次の表に示します:

		p_sep_by_space		
		2	1	0
p_cs_precedes= 1	p_sign_posn= 0	($1.25)	($1.25)	($1.25)
	p_sign_posn= 1	+$1.25	+$1.25	+$1.25
	p_sign_posn= 2	$1.25+	$1.25+	$1.25+
	p_sign_posn= 3	+$1.25	+$1.25	+$1.25
	p_sign_posn= 4	$+1.25	$+1.25	$+1.25
p_cs_precedes= 0	p_sign_posn= 0	(1.25 $)	(1.25 $)	(1.25$)
	p_sign_posn= 1	+1.25 $	+1.25 $	+1.25$
	p_sign_posn= 2	1.25$ +	1.25 $+	1.25$+
	p_sign_posn= 3	1.25+ $	1.25 +$	1.25+$
	p_sign_posn= 4	1.25$ +	1.25 $+	1.25$+

POSIX ロケールでの金額の書式設定定義は次に従います。コードリストは `localedef(1)` の入力を表し、表は同じ情報に `localeconv(3C)` および `nl_langinfo(3C)` 形式を加えたものを表しています。すべての値は POSIX ロケールで未指定です。

```
LC_MONETARY
# This is the POSIX locale definition for
```

```
# the LC_MONETARY category.
#
int_curr_symbol       ""
currency_symbol       ""
mon_decimal_point     ""
mon_thousands_sep     ""
mon_grouping          -1
positive_sign         ""
negative_sign         ""
int_frac_digits       -1
frac_digits           -1
p_cs_precedes         -1
p_sep_by_space        -1
n_cs_precedes         -1
n_sep_by_space        -1
p_sign_posn           -1
n_sign_posn           -1
int_p_cs_precedes     -1
int_p_sep_by_space    -1
int_n_cs_precedes     -1
int_n_sep_by_space    -1
int_p_sign_posn       -1
int_n_sign_posn       -1
#
END LC_MONETARY
```

エントリ n/a は、その値が POSIX ロケールで使用できないことを示します。

LC_NUMERIC

LC_NUMERIC カテゴリは、通貨以外の数値情報の書式設定に使用される規則と記号を定義します。この情報は、localeconv(3C) 関数によって入手できます。

ロケールのこのカテゴリには、次の項目が定義されています。項目名は、ロケールの定義時に localedef ユーティリティーによって認識されるキーワードです。それらは、<locale.h> に定義されている *lconv* 構造のメンバー名にも似ています。localeconv() 関数は、未指定の整数項目に対して {CHAR_MAX} を返し、未指定またはサイズが 0 の文字列項目に対して空の文字列 ("") を返します。

ロケール定義ファイルでは、オペランドは文字列です。一部のキーワードでは、文字列に整数しか含めることができません。提供されていないキーワード、空の文字列 ("") に設定された文字列値、–1 に設定された整数のキーワードは、その値がロケールで使用できないことを示すために使われます。次のキーワードが認識されています:

decimal_point　オペランドは、金額以外の数を書式設定する際に小数点 (基数文字) として使用される記号を含む文字列です。このキーワードを省略することも、空の文字列に設定することもできません。標準

によって decimal_point が単一バイトに制限されているコンテキストでは、複数バイトのオペランドを指定した結果は不確定です。

thousands_sep　オペランドは、書式設定された金額以外の数の中で小数点の左側にある数字の位取り記号として使用される記号を含む文字列です。標準によって thousands_sep が単一バイトに制限されている状況では、複数バイトのオペランドを指定した結果は不確定です。

grouping　書式設定された金額以外の数の中で数字の位取りの桁数を定義します。オペランドは、セミコロンで区切られた一連の整数です。各整数は位取りの桁数を指定します。最初の整数は小数点のすぐ左側の桁数を定義し、以降の整数はその左側の桁数を定義します。最後の整数が –1 でない場合、その前の桁数 (存在する場合) が残りの数字に対して繰り返し使用されます。最後の整数が –1 の場合、これ以上の位取りは行われません。POSIX ロケールでの金額以外の数の書式設定定義は次に従います。コードリストは localedef の入力を表し、表は同じ情報に localeconv 値と nl_langinfo 定数を加えたものを表しています。

```
LC_NUMERIC
# This is the POSIX locale definition for
# the LC_NUMERIC category.
#
decimal_point    "<period>"
thousands_sep    ""
grouping    -1
#
END LC_NUMERIC
```

	POSIX ロケール	langinfo	localeconv()	localedef
項目	値	定数	値	値
decimal_point	"."	RADIXCHAR	"."	.
thousands_sep	n/a	THOUSEP	""	""
grouping	n/a	-	""	–1

エントリ n/a は、その値が POSIX ロケールで使用できないことを示します。

LC_TIME　LC_TIME カテゴリは、date(1) によってサポートされるフィールド記述子の解釈を定義し、strftime(3C)、wcsftime(3C)、strptime(3C)、および nl_langinfo(3C) 関数の動作に影響を及ぼします。C 言語へのアクセスとロケール定義のインタフェースは

かなり異なっているため、それらについては個別に説明します。ロケール定義では、次の必須キーワードが認識されています:

abday　%a フィールド記述子に対応する、曜日の短縮名を定義します (strftime()、wcsftime()、および strptime() 関数の変換仕様)。オペランドは、セミコロンで区切られた 7 つの文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が Sunday に対応する曜日の短縮名で、2 番目が Monday に対応する曜日の短縮名となり、以下同様です。

day　%A フィールド記述子に対応する、曜日の完全な名前を定義します。オペランドは、セミコロンで区切られた 7 つの文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が Sunday に対応する曜日の完全な名前で、2 番目が Monday に対応する曜日の完全な名前となり、以下同様です。

abmon　%b フィールド記述子に対応する、月の短縮名を定義します。オペランドは、セミコロンで区切られた 12 個の文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が 1 年の最初の月 (January) の短縮名で、2 番目が 2 番目の月の短縮名となり、以下同様です。

mon　%B フィールド記述子に対応する、月の完全な名前を定義します。オペランドは、セミコロンで区切られた 12 個の文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が 1 年の最初の月 (January) の完全な名前で、2 番目が 2 番目の月の完全な名前となり、以下同様です。

d_t_fmt　%c フィールド記述子に対応する、日付と時間の適切な表現を定義します。オペランドは 1 つの文字列で構成され、文字とフィールド記述子の任意の組み合わせを含めることができます。また、この文字列にはエスケープシーケンス \\、 \a、\b、\f、\n、\r、\t、\v を含めることもできます。

date_fmt　%C フィールド記述子に対応する、日付と時間の適切な表現を定義します。オペランドは 1 つの文字列で構成され、文字とフィールド記述子の任意の組み合わせを含めることができます。また、この文字列にはエスケープシーケンス \\ 、 \a、\b、\f、\n、\r、\t、\v を含めることもできます。

d_fmt　%x フィールド記述子に対応する、適切な日付表現を定義します。オペランドは 1 つの文字列で構成され、文字とフィールド記述子の任意の組み合わせを含めることができます。また、この文字列にはエスケープシーケンス \\ 、 \a、\b、\f、\n、\r、\t、\v を含めることもできます。

`t_fmt` `%X` フィールド記述子に対応する、適切な時間表現を定義します。オペランドは 1 つの文字列で構成され、文字とフィールド記述子の任意の組み合わせを含めることができます。また、この文字列にはエスケープシーケンス `\\` 、 `\a`、`\b`、`\f`、`\n`、`\r`、`\t`、`\v` を含めることもできます。

`am_pm` `%p` フィールド記述子に対応する、*ante meridiem* および *post meridiem* 文字列の適切な表現を定義します。オペランドは、セミコロンで区切られた 2 つの文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が *ante meridiem* 指定を表し、最後の文字列が *post meridiem* 指定を表します。

`t_fmt_ampm` `%r` フィールド記述子に対応する、`am_pm` を含む 12 時間形式による適切な時間表現を定義します。オペランドは 1 つの文字列で構成され、文字とフィールド記述子の任意の組み合わせを含めることができます。文字列が空の場合、12 時間形式はこのロケールでサポートされません。

`era` ロケールにおける、年号ごとの年の数え方や表示の仕方を定義します。オペランドは、セミコロンで区切られた文字列で構成されます。各文字列は、次の形式による年号記述セグメントです:

direction:*offset*:*start_date*:*end_date*:*era_name*:*era_format*

次の定義に従います。各種年号を記述するのに必要な数だけ年号記述セグメントを指定できます。

年号の開始はもっとも初期の時点ではないことがあります。たとえば、西暦紀元前は西暦元年 1 月 1 日の前日に始まり、時間をさかのぼるにつれて増えていきます。

direction + または – 文字のどちらか。+ 文字は、*start_date* に近い年の方が *end_date* に近い年よりも小さい数字になっていることを示します。– 文字は、*start_date* に近い年の方が *end_date* に近い年よりも大きい数字になっていることを示します。

offset `%Eg` および `%Ey` フィールド記述子に対応する、年号の *start_date* にもっとも近い年を表す数値。

start_date 形式 *yyyy*/*mm*/`dd` による日付。ここでの *yyyy*、*mm*、および `dd` はそれぞれ、年号の開始の年、月、および日を表す数値です。西暦元年より以前の年は、負の数で表されます。

end_date *start_date* と同じ形式、または 2 つの特殊な値 –* または +* のいずれかによる年号の終了日付。–* 値は、最

終日付が時の始まりになることを示します。+* 値は、最終日付が時の終わりになることを示します。

era_name %EC フィールド記述子に対応する、年号の名前を表す文字列。

era_format %EG および %EY フィールド記述子に対応する、年号における年を書式設定するための文字列。

era_d_fmt %Ex フィールド記述子に対応する、代替年号表記による日付の形式を定義します。

era_t_fmt %EX フィールド記述子に対応する、ロケールの適切な代替時間形式を定義します。

era_d_t_fmt %Ec フィールド記述子に対応する、ロケールの適切な代替日付/時間形式を定義します。

alt_digits %O フィールド記述子修飾子に対応する、数字用の代替シンボルを定義します。オペランドは、セミコロンで区切られた文字列で構成され、それぞれが二重引用符で囲まれています。最初の文字列が 0 に対応する代替シンボルで、2 番目の文字列が 1 に対応する代替シンボルとなり、以下同様です。最大 100 個の代替シンボル文字列を指定できます。%O 修飾子は、そのフィールド記述子によって指定された値に対応する文字列がその値の代わりに使用されることを示します。

LC_TIME の *C* 言語へのアクセス

次の情報にアクセスできます。これらは <langinfo.h> に定義されている定数に対応しており、nl_langinfo(3C) 関数の引数として使用できます。

ABDAY_*x* 曜日の短縮名 (Sun など)。*x* は 1 - 7 の数値です。

DAY_*x* 曜日の完全な名前 (Sunday など)。*x* は 1 - 7 の数値です。

ABMON_*x* 月の短縮名 (Jan など)。*x* は 1 - 12 の数値です。

MON_*x* 月の完全な名前 (January など)。*x* は 1 - 12 の数値です。

D_T_FMT 適切な日付と時間の表現。

D_FMT 適切な日付表現。

T_FMT 適切な時間表現。

AM_STR 適切な a.m. 接辞。

PM_STR 適切な p.m. 接辞。

T_FMT_AMPM AM_STR と PM_STR を含む 12 時間形式による適切な時間表現。

ERA

年号記述セグメント。ロケールにおける、年号ごとの年の数え方や表示の仕方を記述します。各年号記述セグメントの形式は次のとおりです:

direction:*offset*:*start_date*:*end_date*:*era_name*:*era_format*

次の定義に従います。各種年号を記述するのに必要な数だけ年号記述セグメントがあります。年号記述セグメントはセミコロンで区切られます。

年号の開始はもっとも初期の時点ではないことがあります。たとえば、西暦紀元前は西暦元年 1 月 1 日の前日に始まり、時間をさかのぼるにつれて増えていきます。

direction
: + または – 文字のどちらか。+ 文字は、*start_date* に近い年の方が *end_date* に近い年よりも小さい数字になっていることを示します。– 文字は、*start_date* に近い年の方が *end_date* に近い年よりも大きい数字になっていることを示します。

offset
: 年号の start_date にもっとも近い年を表す数値。

start_date
: 形式 *yyyy*/*mm*/*dd* による日付。ここでの *yyyy*、*mm*、および dd はそれぞれ、年号の開始の年、月、および日を表す数値です。西暦元年より以前の年は、負の数で表されます。

end_date
: *start_date* と同じ形式、または 2 つの特殊な値 –* または +* のいずれかによる年号の終了日付。–* 値は、最終日付が時の始まりになることを示します。+* 値は、最終日付が時の終わりになることを示します。

era_name
: %EC 変換仕様に対応する年号。

era_format
: %EY および %EY 変換仕様に対応する、年号における年の形式。

ERA_D_FMT
: 年号の日付の形式。

ERA_T_FMT
: %EX フィールド記述子に対応する、ロケールの適切な代替時間形式。

ERA_D_T_FMT
: %Ec フィールド記述子に対応する、ロケールの適切な代替日付/時間形式。

ALT_DIGITS
: %O 変換仕様修飾子に対応する、数字用の代替シンボル。この値はセミコロンで区切られたシンボルで構成されます。最初が 0 に対応する代替シンボルで、2 番目が 1 に対応する代替シンボルとなり、以下同様です。最大 100 個の代替シンボルを指定できます。次の表に、前述の項目間の対応関係

と、date(1)、strftime(3C)、wcsftime(3C)、および strptime(3C) 関数で使用される変換指示子を示します。

localedef キーワード	**langinfo** 定数	変換 指示子
abday	ABDAY_*x*	%a
day	DAY_*x*	%A
abmon	ABMON_*x*	%b
mon	MON	%B
d_t_fmt	D_T_FMT	%c
date_fmt	DATE_FMT	%C
d_fmt	D_FMT	%x
t_fmt	T_FMT	%X
am_pm	AM_STR	%p
am_pm	PM_STR	%p
t_fmt_ampm	T_FMT_AMPM	%r
era	ERA	%EC, %Eg, %EG, %Ey, %EY
era_d_fmt	ERA_D_FMT	%Ex
era_t_fmt	ERA_T_FMT	%EX
era_d_t_fmt	ERA_D_T_FMT	%Ec
alt_digits	ALT_DIGITS	%O

LC_TIME の一般情報

POSIX ロケールのフィールド記述子のいくつか (月の名前など) は頭文字を大文字にして示しますが、ほかのロケールではその必要はありません。これらのフィールドを使用するプログラムでは、出力が文頭に使用される場合に大文字使用の調整が必要になることがあります。

LC_TIME での abday、day、mon、および abmon の説明では、グレゴリオ暦様式のカレンダー (1 週間は 7 日、1 年は 12 か月、うるう年など) を暗に示しています。ほかの種類のカレンダーの時間文字列の書式設定については、このドキュメントでは扱っていません。

「ロケール定義」の date と、strftime(3C) で規定されているように、オプションのキーワードに対応するフィールド記述子は、修飾子とそのあとに続く従来のフィールド記述子 (%Ex など) で構成されています。オプションのキーワードが実装

によってサポートされていなかったり、現在のロケールで未定義だったりする場合、これらのフィールド記述子は従来のフィールド記述子として扱われます。たとえば、次のキーワードがあると想定します:

```
alt_digits    "0th" ; "1st" ; "2nd" ; "3rd" ; "4th" ; "5th" ; \
"6th" ; "7th" ; "8th" ; "9th" ; "10th">
d_fmt    "The %Od day of %B in %Y"
```

7/4/1776 については、%x フィールド記述子によって「The 4th day of July in 1776」となる一方、7/14/1789 は「The 14 day of July in 1789」と表示されます。上記の例は、実例を示すためだけのものです。%0 修飾子は主に、date 形式での漢字またはヒンディー語の数字での使用を意図しています。

LC_MESSAGES

LC_MESSAGES カテゴリは、肯定応答および否定応答の形式と値を定義します。

次のキーワードがロケール定義ファイルの一部として認識されています。nl_langinfo(3C) 関数は、最初の 4 つのキーワードの大文字バージョンを受け入れます。

yesexpr
: オペランドは、肯定または否定応答を求める質問への受け入れ可能な肯定応答を記述する拡張正規表現 (regex(5) を参照) で構成されます。

noexpr
: オペランドは、肯定または否定応答を求める質問への受け入れ可能な否定応答を記述する拡張正規表現で構成されます。

yesstr
: オペランドは、プロンプトなどで受け入れ可能な肯定応答を一覧表示するメッセージを作成するためにアプリケーションで使用できる固定文字列 (正規表現ではなく) で構成されます。

nostr
: オペランドは、受け入れ可能な否定応答を一覧表示するメッセージを作成するためにアプリケーションで使用できる固定文字列で構成されます。POSIX ロケールの肯定および否定応答の形式と値は、localedef の入力を表すコードリスト、同じ情報に nl_langinfo() 定数を加えたものを表す表に従います。

```
LC_MESSAGES
# This is the POSIX locale definition for
# the LC_MESSAGES category.
#
yesexpr "<circumflex><left-square-bracket><y><Y>\
        <right-square-bracket>"
#
noexpr  "<circumflex><left-square-bracket><n><N>\
        <right-square-bracket>"
#
yesstr    "yes"
nostr    "no"
END LC_MESSAGES
```

localedef キーワード	langinfo 定数	POSIX ロケールの値
yesexpr	YESEXPR	"^[yY]"
noexpr	NOEXPR	"^[nN]"
yesstr	YESSTR	"yes"
nostr	NOSTR	"no"

SUSv3 標準に準拠したアプリケーションでは、yesstr および nostr に関する情報は使用できません。

関連項目

date(1), locale(1), localedef(1), sort(1), tr(1), uniq(1), localeconv(3C), nl_langinfo(3C), setlocale(3C), strcoll(3C), strftime(3C), strptime(3C), strxfrm(3C), wcscoll(3C), wcsftime(3C), wcsxfrm(3C), wctype(3C), attributes(5), charmap(5), extensions(5), regex(5)

名前　　locale_alias – ロケール名の別名とそれに対応する標準的なロケール名

機能説明　　システムで受け入れられ、サポートされているロケール名の別名は 2 組あります:

1. setlocale(3C) で指定するようなロケール選択処理
   や、gettext(1)、catopen(3C)、および gettext(3C) で指定するようなメッセージオブジェクトまたはメッセージカタログ処理の実行中に受け入れられ、対応する標準的なロケール名があればその名前にマップされるロケール名の別名。

   マッピング処理中は、可能性のあるマッピングの成功率を上げるために、ロケール名の別名のコードセット名の部分が、英数字のみを抽出して小文字変換することで正規化されます。たとえば、この正規化を使用すると、AR_AA.UTF-8 などのロケール名の別名は、検索ドメイン値として AR_AA.utf8 に正規化されてから、内部マッピングテーブル内の事前に正規化された 1 組のロケール名の別名と実際に比較されて、標準的なロケール名が検索されます。

   マシン順に、サポートされているロケール名の別名を次に示します。対応する標準的なロケールが現時点でシステムにインストールされていないか、または Solaris によってまだサポートされていないために、それらのロケールが現在のシステムに含まれていない可能性があります。

Locale Name Alias	Canonical Locale Name
AR_AE	ar_AE.UTF-8
AR_AE.UTF-8	ar_AE.UTF-8
AR_BH	ar_BH.UTF-8
AR_BH.UTF-8	ar_BH.UTF-8
AR_DZ	ar_DZ.UTF-8
AR_DZ.UTF-8	ar_DZ.UTF-8
AR_EG	ar_EG.UTF-8
AR_EG.UTF-8	ar_EG.UTF-8
AR_JO	ar_JO.UTF-8
AR_JO.UTF-8	ar_JO.UTF-8
AR_KW	ar_KW.UTF-8
AR_KW.UTF-8	ar_KW.UTF-8
AR_MA	ar_MA.UTF-8
AR_MA.UTF-8	ar_MA.UTF-8
AR_OM	ar_OM.UTF-8
AR_OM.UTF-8	ar_OM.UTF-8
AR_QA	ar_QA.UTF-8
AR_QA.UTF-8	ar_QA.UTF-8
AR_SA	ar_SA.UTF-8
AR_SA.UTF-8	ar_SA.UTF-8
AR_TN	ar_TN.UTF-8
AR_TN.UTF-8	ar_TN.UTF-8
AR_YE	ar_YE.UTF-8
AR_YE.UTF-8	ar_YE.UTF-8
AS_IN	as_IN.UTF-8
AS_IN.UTF-8	as_IN.UTF-8

```
AZ_AZ                   az_AZ.UTF-8
AZ_AZ.UTF-8             az_AZ.UTF-8
BE_BY                   be_BY.UTF-8
BE_BY.UTF-8             be_BY.UTF-8
BG_BG                   bg_BG.UTF-8
BG_BG.UTF-8             bg_BG.UTF-8
BN_IN                   bn_IN.UTF-8
BN_IN.UTF-8             bn_IN.UTF-8
CA_ES                   ca_ES.UTF-8
CA_ES.UTF-8             ca_ES.UTF-8
CA_ES.UTF-8@euro        ca_ES.UTF-8
CA_ES@euro              ca_ES.UTF-8
CS_CZ                   cs_CZ.UTF-8
CS_CZ.UTF-8             cs_CZ.UTF-8
DA_DK                   da_DK.UTF-8
DA_DK.UTF-8             da_DK.UTF-8
DE_AT                   de_AT.UTF-8
DE_AT.UTF-8             de_AT.UTF-8
DE_AT.UTF-8@euro        de_AT.UTF-8
DE_AT@euro              de_AT.UTF-8
DE_CH                   de_CH.UTF-8
DE_CH.UTF-8             de_CH.UTF-8
DE_DE                   de_DE.UTF-8
DE_DE.UTF-8             de_DE.UTF-8
DE_DE.UTF-8@euro        de_DE.UTF-8
DE_DE@euro              de_DE.UTF-8
DE_LU                   de_LU.UTF-8
DE_LU.UTF-8             de_LU.UTF-8
DE_LU.UTF-8@euro        de_LU.UTF-8
DE_LU@euro              de_LU.UTF-8
EL_GR                   el_GR.UTF-8
EL_GR.UTF-8             el_GR.UTF-8
EN_AU                   en_AU.UTF-8
EN_AU.UTF-8             en_AU.UTF-8
EN_CA                   en_CA.UTF-8
EN_CA.UTF-8             en_CA.UTF-8
EN_GB                   en_GB.UTF-8
EN_GB.UTF-8             en_GB.UTF-8
EN_GB.UTF-8@euro        en_GB.UTF-8
EN_GB@euro              en_GB.UTF-8
EN_HK                   en_HK.UTF-8
EN_HK.UTF-8             en_HK.UTF-8
EN_IE                   en_IE.UTF-8
EN_IE.UTF-8             en_IE.UTF-8
EN_IE.UTF-8@euro        en_IE.UTF-8
EN_IE@euro              en_IE.UTF-8
EN_IN                   en_IN.UTF-8
EN_IN.UTF-8             en_IN.UTF-8
```

```
EN_NZ                   en_NZ.UTF-8
EN_NZ.UTF-8             en_NZ.UTF-8
EN_PH                   en_PH.UTF-8
EN_PH.UTF-8             en_PH.UTF-8
EN_SG                   en_SG.UTF-8
EN_SG.UTF-8             en_SG.UTF-8
EN_US                   en_US.UTF-8
EN_US.UTF-8             en_US.UTF-8
ES_AR                   es_AR.UTF-8
ES_AR.UTF-8             es_AR.UTF-8
ES_BO                   es_BO.UTF-8
ES_BO.UTF-8             es_BO.UTF-8
ES_CL                   es_CL.UTF-8
ES_CL.UTF-8             es_CL.UTF-8
ES_CO                   es_CO.UTF-8
ES_CO.UTF-8             es_CO.UTF-8
ES_CR                   es_CR.UTF-8
ES_CR.UTF-8             es_CR.UTF-8
ES_DO                   es_DO.UTF-8
ES_DO.UTF-8             es_DO.UTF-8
ES_EC                   es_EC.UTF-8
ES_EC.UTF-8             es_EC.UTF-8
ES_ES                   es_ES.UTF-8
ES_ES.UTF-8             es_ES.UTF-8
ES_ES.UTF-8@euro        es_ES.UTF-8
ES_ES@euro              es_ES.UTF-8
ES_GT                   es_GT.UTF-8
ES_GT.UTF-8             es_GT.UTF-8
ES_HN                   es_HN.UTF-8
ES_HN.UTF-8             es_HN.UTF-8
ES_MX                   es_MX.UTF-8
ES_MX.UTF-8             es_MX.UTF-8
ES_NI                   es_NI.UTF-8
ES_NI.UTF-8             es_NI.UTF-8
ES_PA                   es_PA.UTF-8
ES_PA.UTF-8             es_PA.UTF-8
ES_PE                   es_PE.UTF-8
ES_PE.UTF-8             es_PE.UTF-8
ES_PR                   es_PR.UTF-8
ES_PR.UTF-8             es_PR.UTF-8
ES_PY                   es_PY.UTF-8
ES_PY.UTF-8             es_PY.UTF-8
ES_SV                   es_SV.UTF-8
ES_SV.UTF-8             es_SV.UTF-8
ES_US                   es_US.UTF-8
ES_US.UTF-8             es_US.UTF-8
ES_UY                   es_UY.UTF-8
ES_UY.UTF-8             es_UY.UTF-8
```

```
ES_VE                   es_VE.UTF-8
ES_VE.UTF-8             es_VE.UTF-8
ET_EE                   et_EE.UTF-8
ET_EE.UTF-8             et_EE.UTF-8
FI_FI                   fi_FI.UTF-8
FI_FI.UTF-8             fi_FI.UTF-8
FI_FI.UTF-8@euro        fi_FI.UTF-8
FI_FI@euro              fi_FI.UTF-8
FR_BE                   fr_BE.UTF-8
FR_BE.UTF-8             fr_BE.UTF-8
FR_BE.UTF-8@euro        fr_BE.UTF-8
FR_BE@euro              fr_BE.UTF-8
FR_CA                   fr_CA.UTF-8
FR_CA.UTF-8             fr_CA.UTF-8
FR_CH                   fr_CH.UTF-8
FR_CH.UTF-8             fr_CH.UTF-8
FR_FR                   fr_FR.UTF-8
FR_FR.UTF-8             fr_FR.UTF-8
FR_FR.UTF-8@euro        fr_FR.UTF-8
FR_FR@euro              fr_FR.UTF-8
FR_LU                   fr_LU.UTF-8
FR_LU.UTF-8             fr_LU.UTF-8
FR_LU.UTF-8@euro        fr_LU.UTF-8
FR_LU@euro              fr_LU.UTF-8
GU_IN                   gu_IN.UTF-8
GU_IN.UTF-8             gu_IN.UTF-8
HE_IL                   he_IL.UTF-8
HE_IL.UTF-8             he_IL.UTF-8
HI_IN                   hi_IN.UTF-8
HI_IN.UTF-8             hi_IN.UTF-8
HR_HR                   hr_HR.UTF-8
HR_HR.UTF-8             hr_HR.UTF-8
HU_HU                   hu_HU.UTF-8
HU_HU.UTF-8             hu_HU.UTF-8
ID_ID                   id_ID.UTF-8
ID_ID.UTF-8             id_ID.UTF-8
IS_IS                   is_IS.UTF-8
IS_IS.UTF-8             is_IS.UTF-8
IT_CH                   it_CH.UTF-8
IT_CH.UTF-8             it_CH.UTF-8
IT_IT                   it_IT.UTF-8
IT_IT.UTF-8             it_IT.UTF-8
IT_IT.UTF-8@euro        it_IT.UTF-8
IT_IT@euro              it_IT.UTF-8
JA_JP                   ja_JP.UTF-8
JA_JP.UTF-8             ja_JP.UTF-8
KK_KZ                   kk_KZ.UTF-8
KK_KZ.UTF-8             kk_KZ.UTF-8
```

```
KN_IN                   kn_IN.UTF-8
KN_IN.UTF-8             kn_IN.UTF-8
KO_KR                   ko_KR.UTF-8
KO_KR.UTF-8             ko_KR.UTF-8
LT_LT                   lt_LT.UTF-8
LT_LT.UTF-8             lt_LT.UTF-8
LV_LV                   lv_LV.UTF-8
LV_LV.UTF-8             lv_LV.UTF-8
MK_MK                   mk_MK.UTF-8
MK_MK.UTF-8             mk_MK.UTF-8
ML_IN                   ml_IN.UTF-8
ML_IN.UTF-8             ml_IN.UTF-8
MR_IN                   mr_IN.UTF-8
MR_IN.UTF-8             mr_IN.UTF-8
MS_MY                   ms_MY.UTF-8
MS_MY.UTF-8             ms_MY.UTF-8
MT_MT                   mt_MT.UTF-8
MT_MT.UTF-8             mt_MT.UTF-8
NL_BE                   nl_BE.UTF-8
NL_BE.UTF-8             nl_BE.UTF-8
NL_BE.UTF-8@euro        nl_BE.UTF-8
NL_BE@euro              nl_BE.UTF-8
NL_NL                   nl_NL.UTF-8
NL_NL.UTF-8             nl_NL.UTF-8
NL_NL.UTF-8@euro        nl_NL.UTF-8
NL_NL@euro              nl_NL.UTF-8
OR_IN                   or_IN.UTF-8
OR_IN.UTF-8             or_IN.UTF-8
PA_IN                   pa_IN.UTF-8
PA_IN.UTF-8             pa_IN.UTF-8
PL_PL                   pl_PL.UTF-8
PL_PL.UTF-8             pl_PL.UTF-8
PT_BR                   pt_BR.UTF-8
PT_BR.UTF-8             pt_BR.UTF-8
PT_PT                   pt_PT.UTF-8
PT_PT.UTF-8             pt_PT.UTF-8
PT_PT.UTF-8@euro        pt_PT.UTF-8
PT_PT@euro              pt_PT.UTF-8
RO_RO                   ro_RO.UTF-8
RO_RO.UTF-8             ro_RO.UTF-8
RU_RU                   ru_RU.UTF-8
RU_RU.UTF-8             ru_RU.UTF-8
SK_SK                   sk_SK.UTF-8
SK_SK.UTF-8             sk_SK.UTF-8
SL_SI                   sl_SI.UTF-8
SL_SI.UTF-8             sl_SI.UTF-8
SQ_AL                   sq_AL.UTF-8
SQ_AL.UTF-8             sq_AL.UTF-8
```

```
SV_SE                   sv_SE.UTF-8
SV_SE.UTF-8             sv_SE.UTF-8
TA_IN                   ta_IN.UTF-8
TA_IN.UTF-8             ta_IN.UTF-8
TE_IN                   te_IN.UTF-8
TE_IN.UTF-8             te_IN.UTF-8
TH_TH                   th_TH.UTF-8
TH_TH.UTF-8             th_TH.UTF-8
TR_TR                   tr_TR.UTF-8
TR_TR.UTF-8             tr_TR.UTF-8
UK_UA                   uk_UA.UTF-8
UK_UA.UTF-8             uk_UA.UTF-8
VI_VN                   vi_VN.UTF-8
VI_VN.UTF-8             vi_VN.UTF-8
ZH_CN                   zh_CN.UTF-8
ZH_CN.UTF-8             zh_CN.UTF-8
ZH_HK                   zh_HK.UTF-8
ZH_HK.UTF-8             zh_HK.UTF-8
ZH_SG                   zh_SG.UTF-8
ZH_SG.UTF-8             zh_SG.UTF-8
ZH_TW                   zh_TW.UTF-8
ZH_TW.UTF-8             zh_TW.UTF-8
Zh_CN                   zh_CN.GB18030
Zh_CN.GB18030           zh_CN.GB18030
Zh_TW                   zh_TW.BIG5
Zh_TW.big5              zh_TW.BIG5
af_ZA.utf8              af_ZA.UTF-8
ar                      ar_EG.ISO8859-6
ar_AE.utf8              ar_AE.UTF-8
ar_BH.utf8              ar_BH.UTF-8
ar_DZ.utf8              ar_DZ.UTF-8
ar_EG                   ar_EG.ISO8859-6
ar_EG.iso88596          ar_EG.ISO8859-6
ar_EG.utf8              ar_EG.UTF-8
ar_IQ.utf8              ar_IQ.UTF-8
ar_JO.utf8              ar_JO.UTF-8
ar_KW.utf8              ar_KW.UTF-8
ar_LY.utf8              ar_LY.UTF-8
ar_MA.utf8              ar_MA.UTF-8
ar_OM.utf8              ar_OM.UTF-8
ar_QA.utf8              ar_QA.UTF-8
ar_SA.utf8              ar_SA.UTF-8
ar_TN.utf8              ar_TN.UTF-8
ar_YE.utf8              ar_YE.UTF-8
as_IN.utf8              as_IN.UTF-8
az_AZ.utf8              az_AZ.UTF-8
be_BY.utf8              be_BY.UTF-8
bg_BG                   bg_BG.ISO8859-5
```

```
bg_BG.utf8              bg_BG.UTF-8
bn_IN                   bn_IN.UTF-8
bn_IN.utf8              bn_IN.UTF-8
bs_BA                   bs_BA.ISO8859-2
bs_BA.iso88592          bs_BA.ISO8859-2
bs_BA.utf8              bs_BA.UTF-8
ca                      ca_ES.ISO8859-1
ca_ES                   ca_ES.ISO8859-1
ca_ES.8859-15           ca_ES.ISO8859-15
ca_ES.8859-15@euro      ca_ES.ISO8859-15
ca_ES.ISO8859-15@euro   ca_ES.ISO8859-15
ca_ES.iso88591          ca_ES.ISO8859-1
ca_ES.utf8              ca_ES.UTF-8
ca_ES@euro              ca_ES.ISO8859-15
cs                      cs_CZ.ISO8859-2
cs_CZ                   cs_CZ.ISO8859-2
cs_CZ.iso88592          cs_CZ.ISO8859-2
cs_CZ.utf8              cs_CZ.UTF-8
da                      da_DK.ISO8859-1
da.ISO8859-15           da_DK.ISO8859-15
da_DK                   da_DK.ISO8859-1
da_DK.8859-15           da_DK.ISO8859-15
da_DK.iso88591          da_DK.ISO8859-1
da_DK.iso885915         da_DK.ISO8859-15
da_DK.iso885915@euro    da_DK.ISO8859-15
da_DK.utf8              da_DK.UTF-8
de                      de_DE.ISO8859-1
de.ISO8859-15           de_DE.ISO8859-15
de.UTF-8                de_DE.UTF-8
de_AT                   de_AT.ISO8859-1
de_AT.8859-15           de_AT.ISO8859-15
de_AT.8859-15@euro      de_AT.ISO8859-15
de_AT.ISO8859-15@euro   de_AT.ISO8859-15
de_AT.iso88591          de_AT.ISO8859-1
de_AT.utf8              de_AT.UTF-8
de_AT@euro              de_AT.ISO8859-15
de_BE.utf8              de_BE.UTF-8
de_CH                   de_CH.ISO8859-1
de_CH.iso88591          de_CH.ISO8859-1
de_CH.utf8              de_CH.UTF-8
de_DE                   de_DE.ISO8859-1
de_DE.8859-15           de_DE.ISO8859-15
de_DE.8859-15@euro      de_DE.ISO8859-15
de_DE.ISO8859-15@euro   de_DE.ISO8859-15
de_DE.UTF-8@euro        de_DE.UTF-8
de_DE.iso88591          de_DE.ISO8859-1
de_DE.utf8              de_DE.UTF-8
de_DE@euro              de_DE.ISO8859-15
```

```
de_LU.utf8              de_LU.UTF-8
el                      el_GR.ISO8859-7
el.UTF-8                el_CY.UTF-8
el.sun_eu_greek         el_GR.ISO8859-7
el_CY.utf8              el_CY.UTF-8
el_GR                   el_GR.ISO8859-7
el_GR.ISO8859-7@euro    el_GR.ISO8859-7
el_GR.iso88597          el_GR.ISO8859-7
el_GR.utf8              el_GR.UTF-8
en_AU                   en_AU.ISO8859-1
en_AU.iso88591          en_AU.ISO8859-1
en_AU.utf8              en_AU.UTF-8
en_BW.utf8              en_BW.UTF-8
en_CA                   en_CA.ISO8859-1
en_CA.iso88591          en_CA.ISO8859-1
en_CA.utf8              en_CA.UTF-8
en_GB                   en_GB.ISO8859-1
en_GB.8859-15@euro      en_GB.ISO8859-15
en_GB.IBM-1252          en_GB.ANSI1252
en_GB.iso88591          en_GB.ISO8859-1
en_GB.iso885915         en_GB.ISO8859-15
en_GB.iso885915@euro    en_GB.ISO8859-15
en_GB.utf8              en_GB.UTF-8
en_HK.utf8              en_HK.UTF-8
en_IE                   en_IE.ISO8859-1
en_IE.8859-15           en_IE.ISO8859-15
en_IE.8859-15@euro      en_IE.ISO8859-15
en_IE.ISO8859-15@euro   en_IE.ISO8859-15
en_IE.iso88591          en_IE.ISO8859-1
en_IE.utf8              en_IE.UTF-8
en_IE@euro              en_IE.ISO8859-15
en_IN.utf8              en_IN.UTF-8
en_NZ                   en_NZ.ISO8859-1
en_NZ.iso88591          en_NZ.ISO8859-1
en_NZ.utf8              en_NZ.UTF-8
en_PH.utf8              en_PH.UTF-8
en_SG.utf8              en_SG.UTF-8
en_US                   en_US.ISO8859-1
en_US.8859-15           en_US.ISO8859-15
en_US.iso88591          en_US.ISO8859-1
en_US.iso885915         en_US.ISO8859-15
en_US.utf8              en_US.UTF-8
en_ZW.utf8              en_ZW.UTF-8
es                      es_ES.ISO8859-1
es.ISO8859-15           es_ES.ISO8859-15
es.UTF-8                es_ES.UTF-8
es_AR                   es_AR.ISO8859-1
es_AR.iso88591          es_AR.ISO8859-1
```

```
es_AR.utf8              es_AR.UTF-8
es_BO                   es_BO.ISO8859-1
es_BO.iso88591          es_BO.ISO8859-1
es_BO.utf8              es_BO.UTF-8
es_CL                   es_CL.ISO8859-1
es_CL.iso88591          es_CL.ISO8859-1
es_CL.utf8              es_CL.UTF-8
es_CO                   es_CO.ISO8859-1
es_CO.iso88591          es_CO.ISO8859-1
es_CO.utf8              es_CO.UTF-8
es_CR                   es_CR.ISO8859-1
es_CR.iso88591          es_CR.ISO8859-1
es_CR.utf8              es_CR.UTF-8
es_DO.utf8              es_DO.UTF-8
es_EC                   es_EC.ISO8859-1
es_EC.iso88591          es_EC.ISO8859-1
es_EC.utf8              es_EC.UTF-8
es_ES                   es_ES.ISO8859-1
es_ES.8859-15           es_ES.ISO8859-15
es_ES.8859-15@euro      es_ES.ISO8859-15
es_ES.ISO8859-15@euro   es_ES.ISO8859-15
es_ES.UTF-8@euro        es_ES.UTF-8
es_ES.iso88591          es_ES.ISO8859-1
es_ES.utf8              es_ES.UTF-8
es_ES@euro              es_ES.ISO8859-15
es_GT                   es_GT.ISO8859-1
es_GT.iso88591          es_GT.ISO8859-1
es_GT.utf8              es_GT.UTF-8
es_HN.utf8              es_HN.UTF-8
es_MX                   es_MX.ISO8859-1
es_MX.iso88591          es_MX.ISO8859-1
es_NI.utf8              es_NI.UTF-8
es_PA                   es_PA.ISO8859-1
es_PA.iso88591          es_PA.ISO8859-1
es_PA.utf8              es_PA.UTF-8
es_PE                   es_PE.ISO8859-1
es_PE.iso88591          es_PE.ISO8859-1
es_PE.utf8              es_PE.UTF-8
es_PR.utf8              es_PR.UTF-8
es_PY                   es_PY.ISO8859-1
es_PY.iso88591          es_PY.ISO8859-1
es_PY.utf8              es_PY.UTF-8
es_SV                   es_SV.ISO8859-1
es_SV.iso88591          es_SV.ISO8859-1
es_SV.utf8              es_SV.UTF-8
es_US.utf8              es_US.UTF-8
es_UY                   es_UY.ISO8859-1
es_UY.8859-15           es_UY.ISO8859-15
```

```
es_UY.iso88591          es_UY.ISO8859-1
es_UY.utf8              es_UY.UTF-8
es_VE                   es_VE.ISO8859-1
es_VE.iso88591          es_VE.ISO8859-1
es_VE.utf8              es_VE.UTF-8
et                      et_EE.ISO8859-15
et_EE                   et_EE.ISO8859-15
et_EE.iso885915         et_EE.ISO8859-15
et_EE.utf8              et_EE.UTF-8
fi                      fi_FI.ISO8859-1
fi.ISO8859-15           fi_FI.ISO8859-15
fi_FI                   fi_FI.ISO8859-1
fi_FI.8859-15           fi_FI.ISO8859-15
fi_FI.8859-15@euro      fi_FI.ISO8859-15
fi_FI.ISO8859-15@euro   fi_FI.ISO8859-15
fi_FI.iso88591          fi_FI.ISO8859-1
fi_FI.utf8              fi_FI.UTF-8
fi_FI@euro              fi_FI.ISO8859-15
fr                      fr_FR.ISO8859-1
fr.ISO8859-15           fr_FR.ISO8859-15
fr.UTF-8                fr_FR.UTF-8
fr_BE                   fr_BE.ISO8859-1
fr_BE.8859-15           fr_BE.ISO8859-15
fr_BE.8859-15@euro      fr_BE.ISO8859-15
fr_BE.ISO8859-15@euro   fr_BE.ISO8859-15
fr_BE.UTF-8@euro        fr_BE.UTF-8
fr_BE.iso88591          fr_BE.ISO8859-1
fr_BE.utf8              fr_BE.UTF-8
fr_BE@euro              fr_BE.ISO8859-15
fr_CA                   fr_CA.ISO8859-1
fr_CA.iso88591          fr_CA.ISO8859-1
fr_CA.utf8              fr_CA.UTF-8
fr_CH                   fr_CH.ISO8859-1
fr_CH.iso88591          fr_CH.ISO8859-1
fr_CH.utf8              fr_CH.UTF-8
fr_FR                   fr_FR.ISO8859-1
fr_FR.8859-15           fr_FR.ISO8859-15
fr_FR.8859-15@euro      fr_FR.ISO8859-15
fr_FR.ISO8859-15@euro   fr_FR.ISO8859-15
fr_FR.UTF-8@euro        fr_FR.UTF-8
fr_FR.iso88591          fr_FR.ISO8859-1
fr_FR.utf8              fr_FR.UTF-8
fr_FR@euro              fr_FR.ISO8859-15
fr_LU.utf8              fr_LU.UTF-8
fr_LU@euro              fr_LU.ISO8859-15
gu_IN                   gu_IN.UTF-8
gu_IN.utf8              gu_IN.UTF-8
he                      he_IL.ISO8859-8
```

he_IL	he_IL.ISO8859-8
he_IL.iso88598	he_IL.ISO8859-8
he_IL.utf8	he_IL.UTF-8
hi_IN	hi_IN.UTF-8
hi_IN.utf8	hi_IN.UTF-8
hr_HR	hr_HR.ISO8859-2
hr_HR.iso88592	hr_HR.ISO8859-2
hr_HR.utf8	hr_HR.UTF-8
hu	hu_HU.ISO8859-2
hu_HU	hu_HU.ISO8859-2
hu_HU.iso88592	hu_HU.ISO8859-2
hu_HU.utf8	hu_HU.UTF-8
hy_AM	hy_AM.UTF-8
hy_AM.utf8	hy_AM.UTF-8
id_ID.utf8	id_ID.UTF-8
is_IS	is_IS.ISO8859-1
is_IS.iso88591	is_IS.ISO8859-1
is_IS.utf8	is_IS.UTF-8
it	it_IT.ISO8859-1
it.ISO8859-15	it_IT.ISO8859-15
it.UTF-8	it_IT.UTF-8
it_CH.utf8	it_CH.UTF-8
it_IT	it_IT.ISO8859-1
it_IT.8859-15	it_IT.ISO8859-15
it_IT.8859-15@euro	it_IT.ISO8859-15
it_IT.ISO8859-15@euro	it_IT.ISO8859-15
it_IT.UTF-8@euro	it_IT.UTF-8
it_IT.iso88591	it_IT.ISO8859-1
it_IT.utf8	it_IT.UTF-8
it_IT@euro	it_IT.ISO8859-15
ja	ja_JP.eucJP
ja_JP	ja_JP.eucJP
ja_JP.IBM-eucJP	ja_JP.eucJP
ja_JP.eucjp	ja_JP.eucJP
ja_JP.utf8	ja_JP.UTF-8
ka_GE.utf8	ka_GE.UTF-8
kk_KZ.utf8	kk_KZ.UTF-8
kn_IN	kn_IN.UTF-8
kn_IN.utf8	kn_IN.UTF-8
ko	ko_KR.EUC
ko.UTF-8	ko_KR.UTF-8
ko_KR	ko_KR.EUC
ko_KR.IBM-eucKR	ko_KR.EUC
ko_KR.euckr	ko_KR.EUC
ko_KR.utf8	ko_KR.UTF-8
ku_TR.utf8	ku_TR.UTF-8
ky_KG	ky_KG.UTF-8
ky_KG.utf8	ky_KG.UTF-8

```
lt                      lt_LT.ISO8859-13
lt_LT                   lt_LT.ISO8859-13
lt_LT.iso885913         lt_LT.ISO8859-13
lt_LT.utf8              lt_LT.UTF-8
lv                      lv_LV.ISO8859-13
lv_LV                   lv_LV.ISO8859-13
lv_LV.iso885913         lv_LV.ISO8859-13
lv_LV.utf8              lv_LV.UTF-8
mk_MK                   mk_MK.ISO8859-5
mk_MK.iso88595          mk_MK.ISO8859-5
mk_MK.utf8              mk_MK.UTF-8
ml_IN                   ml_IN.UTF-8
ml_IN.utf8              ml_IN.UTF-8
mr_IN                   mr_IN.UTF-8
mr_IN.utf8              mr_IN.UTF-8
ms_MY.utf8              ms_MY.UTF-8
mt_MT.utf8              mt_MT.UTF-8
nb_NO                   nb_NO.ISO8859-1
nb_NO.iso88591          nb_NO.ISO8859-1
nb_NO.utf8              nb_NO.UTF-8
nl                      nl_NL.ISO8859-1
nl.ISO8859-15           nl_NL.ISO8859-15
nl_BE                   nl_BE.ISO8859-1
nl_BE.8859-15           nl_BE.ISO8859-15
nl_BE.8859-15@euro      nl_BE.ISO8859-15
nl_BE.ISO8859-15@euro   nl_BE.ISO8859-15
nl_BE.iso88591          nl_BE.ISO8859-1
nl_BE.utf8              nl_BE.UTF-8
nl_BE@euro              nl_BE.ISO8859-15
nl_NL                   nl_NL.ISO8859-1
nl_NL.8859-15           nl_NL.ISO8859-15
nl_NL.8859-15@euro      nl_NL.ISO8859-15
nl_NL.ISO8859-15@euro   nl_NL.ISO8859-15
nl_NL.iso88591          nl_NL.ISO8859-1
nl_NL.utf8              nl_NL.UTF-8
nl_NL@euro              nl_NL.ISO8859-15
nn_NO                   nn_NO.ISO8859-1
nn_NO.iso88591          nn_NO.ISO8859-1
nn_NO.utf8              nn_NO.UTF-8
no                      nb_NO.ISO8859-1
no_NO                   nb_NO.ISO8859-1
no_NO.ISO8859-1@bokmal  nb_NO.ISO8859-1
no_NO.ISO8859-1@nynorsk nn_NO.ISO8859-1
no_NY                   nn_NO.ISO8859-1
or_IN                   or_IN.UTF-8
or_IN.utf8              or_IN.UTF-8
pa_IN                   pa_IN.UTF-8
pa_IN.utf8              pa_IN.UTF-8
```

pl	pl_PL.ISO8859-2
pl.UTF-8	pl_PL.UTF-8
pl_PL	pl_PL.ISO8859-2
pl_PL.iso88592	pl_PL.ISO8859-2
pl_PL.utf8	pl_PL.UTF-8
pt	pt_PT.ISO8859-1
pt.ISO8859-15	pt_PT.ISO8859-15
pt_BR	pt_BR.ISO8859-1
pt_BR.iso88591	pt_BR.ISO8859-1
pt_BR.utf8	pt_BR.UTF-8
pt_PT	pt_PT.ISO8859-1
pt_PT.8859-15	pt_PT.ISO8859-15
pt_PT.8859-15@euro	pt_PT.ISO8859-15
pt_PT.ISO8859-15@euro	pt_PT.ISO8859-15
pt_PT.iso88591	pt_PT.ISO8859-1
pt_PT.utf8	pt_PT.UTF-8
pt_PT@euro	pt_PT.ISO8859-15
ro_RO	ro_RO.ISO8859-2
ro_RO.iso88592	ro_RO.ISO8859-2
ro_RO.utf8	ro_RO.UTF-8
ru	ru_RU.ISO8859-5
ru.UTF-8	ru_RU.UTF-8
ru.koi8-r	ru_RU.KOI8-R
ru_RU	ru_RU.ISO8859-5
ru_RU.iso88595	ru_RU.ISO8859-5
ru_RU.koi8r	ru_RU.KOI8-R
ru_RU.utf8	ru_RU.UTF-8
ru_UA.utf8	ru_UA.UTF-8
sh	bs_BA.ISO8859-2
sh_BA	bs_BA.ISO8859-2
sh_BA.ISO8859-2@bosnia	bs_BA.ISO8859-2
sh_BA.UTF-8	bs_BA.UTF-8
sk_SK	sk_SK.ISO8859-2
sk_SK.iso88592	sk_SK.ISO8859-2
sk_SK.utf8	sk_SK.UTF-8
sl_SI	sl_SI.ISO8859-2
sl_SI.iso88592	sl_SI.ISO8859-2
sl_SI.utf8	sl_SI.UTF-8
sq_AL	sq_AL.ISO8859-2
sq_AL.utf8	sq_AL.UTF-8
sr_CS.UTF-8	sr_RS.UTF-8
sr_CS.iso88595	sr_CS.ISO8859-5
sr_ME	sr_ME.UTF-8
sr_ME.utf8	sr_ME.UTF-8
sr_RS	sr_RS.UTF-8
sr_RS.utf8	sr_RS.UTF-8
sv	sv_SE.ISO8859-1
sv.ISO8859-15	sv_SE.ISO8859-15

```
sv.UTF-8                sv_SE.UTF-8
sv_SE                   sv_SE.ISO8859-1
sv_SE.8859-15           sv_SE.ISO8859-15
sv_SE.iso88591          sv_SE.ISO8859-1
sv_SE.iso885915         sv_SE.ISO8859-15
sv_SE.iso885915@euro    sv_SE.ISO8859-15
sv_SE.utf8              sv_SE.UTF-8
ta_IN                   ta_IN.UTF-8
ta_IN.utf8              ta_IN.UTF-8
te_IN                   te_IN.UTF-8
te_IN.utf8              te_IN.UTF-8
th                      th_TH.TIS620
th_TH                   th_TH.TIS620
th_TH.ISO8859-11        th_TH.TIS620
th_TH.TIS-620           th_TH.TIS620
th_TH.utf8              th_TH.UTF-8
tr                      tr_TR.ISO8859-9
tr_TR                   tr_TR.ISO8859-9
tr_TR.iso88599          tr_TR.ISO8859-9
tr_TR.utf8              tr_TR.UTF-8
uk_UA.utf8              uk_UA.UTF-8
vi_VN                   vi_VN.UTF-8
vi_VN.utf8              vi_VN.UTF-8
zh                      zh_CN.EUC
zh.GBK                  zh_CN.GBK
zh.UTF-8                zh_CN.UTF-8
zh_CN                   zh_CN.EUC
zh_CN.IBM-eucCN         zh_CN.EUC
zh_CN.gb18030           zh_CN.GB18030
zh_CN.gb2312            zh_CN.EUC
zh_CN.gbk               zh_CN.GBK
zh_CN.utf8              zh_CN.UTF-8
zh_HK                   zh_HK.BIG5HK
zh_HK.big5hkscs         zh_HK.BIG5HK
zh_HK.utf8              zh_HK.UTF-8
zh_SG.utf8              zh_SG.UTF-8
zh_TW                   zh_TW.EUC
zh_TW.IBM-eucTW         zh_TW.EUC
zh_TW.big5              zh_TW.BIG5
zh_TW.euctw             zh_TW.EUC
zh_TW.utf8              zh_TW.UTF-8
```

2. gettext(1)、catopen(3C)、および gettext(3C) で指定するようなメッセージオブジェクトまたはメッセージカタログファイルの検索のためにさらに照合される、廃止された Solaris ロケール名

   現在のロケールが記載されている標準的なロケールのいずれかであるときに、現在実行中のプログラムで現在のロケール名を使って開く、対応するメッセージオブジェクトまたはメッセージカタログファイルが存在しない場

合、メッセージ処理関数では次に示す追加のロケール名を別名として使用することで、開くメッセージオブジェクトまたはメッセージカタログファイルの存在をさらに調べます:

```
Canonical Locale Name    Additional Locale Names Checked
---------------------    -------------------------------
ar_EG.ISO8859-6          ar
bg_BG.ISO8859-5          bg_BG
bs_BA.ISO8859-2          sh, sh_BA, sh_BA.ISO8859-2@bosnia
bs_BA.UTF-8              sh_BA.UTF-8
ca_ES.ISO8859-1          ca, ca_ES
ca_ES.ISO8859-15         ca_ES.ISO8859-15@euro
cs_CZ.ISO8859-2          cs, cs_CZ
da_DK.ISO8859-1          da, da_DK
da_DK.ISO8859-15         da.ISO8859-15
de_AT.ISO8859-1          de_AT
de_AT.ISO8859-15         de_AT.ISO8859-15@euro
de_CH.ISO8859-1          de_CH
de_DE.ISO8859-1          de, de_DE
de_DE.ISO8859-15         de.ISO8859-15, de_DE.ISO8859-15@euro
de_DE.UTF-8              de.UTF-8, de_DE.UTF-8@euro
el_CY.UTF-8              el.UTF-8
el_GR.ISO8859-7          el, el.sun_eu_greek, el_GR,
                         el_GR.ISO8859-7@euro
en_AU.ISO8859-1          en_AU
en_CA.ISO8859-1          en_CA
en_GB.ISO8859-1          en_GB
en_IE.ISO8859-1          en_IE
en_IE.ISO8859-15         en_IE.ISO8859-15@euro
en_NZ.ISO8859-1          en_NZ
en_US.ISO8859-1          en_US
es_AR.ISO8859-1          es_AR
es_BO.ISO8859-1          es_BO
es_CL.ISO8859-1          es_CL
es_CO.ISO8859-1          es_CO
es_CR.ISO8859-1          es_CR
es_EC.ISO8859-1          es_EC
es_ES.ISO8859-1          es, es_ES
es_ES.ISO8859-15         es.ISO8859-15, es_ES.ISO8859-15@euro
es_ES.UTF-8              es.UTF-8, es_ES.UTF-8@euro
es_GT.ISO8859-1          es_GT
es_MX.ISO8859-1          es_MX
es_NI.ISO8859-1          es_NI
es_PA.ISO8859-1          es_PA
es_PE.ISO8859-1          es_PE
es_PY.ISO8859-1          es_PY
es_SV.ISO8859-1          es_SV
es_UY.ISO8859-1          es_UY
```

```
es_VE.ISO8859-1         es_VE
et_EE.ISO8859-15        et, et_EE
fi_FI.ISO8859-1         fi, fi_FI
fi_FI.ISO8859-15        fi.ISO8859-15, fi_FI.ISO8859-15@euro
fr_BE.ISO8859-1         fr_BE
fr_BE.ISO8859-15        fr_BE.ISO8859-15@euro
fr_BE.UTF-8             fr_BE.UTF-8@euro
fr_CA.ISO8859-1         fr_CA
fr_CH.ISO8859-1         fr_CH
fr_FR.ISO8859-1         fr, fr_FR
fr_FR.ISO8859-15        fr.ISO8859-15, fr_FR.ISO8859-15@euro
fr_FR.UTF-8             fr.UTF-8, fr_FR.UTF-8@euro
he_IL.ISO8859-8         he, he_IL
hr_HR.ISO8859-2         hr_HR
hu_HU.ISO8859-2         hu, hu_HU
is_IS.ISO8859-1         is_IS
it_IT.ISO8859-1         it, it_IT
it_IT.ISO8859-15        it.ISO8859-15, it_IT.ISO8859-15@euro
it_IT.UTF-8             it.UTF-8, it_IT.UTF-8@euro
ja_JP.eucJP             ja
ko_KR.EUC               ko
ko_KR.UTF-8             ko.UTF-8
lt_LT.ISO8859-13        lt, lt_LT
lv_LV.ISO8859-13        lv, lv_LV
mk_MK.ISO8859-5         mk_MK
nb_NO.ISO8859-1         no, no_NO, no_NO.ISO8859-1@bokmal
nl_BE.ISO8859-1         nl_BE
nl_BE.ISO8859-15        nl_BE.ISO8859-15@euro
nl_NL.ISO8859-1         nl, nl_NL
nl_NL.ISO8859-15        nl.ISO8859-15, nl_NL.ISO8859-15@euro
nn_NO.ISO8859-1         no_NO.ISO8859-1@nynorsk, no_NY
pl_PL.ISO8859-2         pl, pl_PL
pl_PL.UTF-8             pl.UTF-8
pt_BR.ISO8859-1         pt_BR
pt_PT.ISO8859-1         pt, pt_PT
pt_PT.ISO8859-15        pt.ISO8859-15, pt_PT.ISO8859-15@euro
ro_RO.ISO8859-2         ro_RO
ru_RU.ISO8859-5         ru, ru_RU
ru_RU.KOI8-R            ru.koi8-r
ru_RU.UTF-8             ru.UTF-8
sk_SK.ISO8859-2         sk_SK
sl_SI.ISO8859-2         sl_SI
sq_AL.ISO8859-2         sq_AL
sr_ME.ISO8859-5         sr_SP, sr_YU, sr_YU.ISO8859-5
sr_ME.UTF-8             sr_CS, sr_CS.UTF-8
sr_RS.ISO8859-5         sr_SP, sr_YU, sr_YU.ISO8859-5
sr_RS.UTF-8             sr_CS, sr_CS.UTF-8
sv_SE.ISO8859-1         sv, sv_SE
```

```
sv_SE.ISO8859-15        sv.ISO8859-15
sv_SE.UTF-8             sv.UTF-8
th_TH.TIS620            th, th_TH, th_TH.ISO8859-11
tr_TR.ISO8859-9         tr, tr_TR
zh_CN.EUC               zh
zh_CN.GBK               zh.GBK
zh_CN.UTF-8             zh.UTF-8
zh_TW.EUC               zh_TW
```

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実

関連項目

gettext(1)、catopen(3C)、gettext(3C)、setlocale(3C)、attributes(5)、environ(5)

名前

man – macros to format Reference Manual pages

形式

`nroff -man` *filename*`...`

`troff -man` *filename*`...`

機能説明

These macros are used to lay out the reference pages in this manual. If *filename* contains format input for a preprocessor, the commands shown above must be piped through the appropriate preprocessor. This is handled automatically by the `man(1)` command. See the `Conventions` section.

Any text argument *t* may be zero to six words. Quotes may be used to include SPACE characters in a "word". If *text* is empty, the special treatment is applied to the next input line with text to be printed. In this way `.I` may be used to italicize a whole line, or `.SB` may be used to make small bold letters.

A prevailing indent distance is remembered between successive indented paragraphs, and is reset to default value upon reaching a non-indented paragraph. Default units for indents *i* are ens.

Type font and size are reset to default values before each paragraph, and after processing font and size setting macros.

These strings are predefined by `-man`:

`\*R` '®', '(Reg)' in `nroff`.

`\*S` Change to default type size.

Requests

* n.t.l. = next text line; p.i. = prevailing indent

Request	*Cause Break*	*If no Argument*	*Explanation*
`.B`*t*	no	*t*=n.t.l.*	Text is in bold font.
`.BI`*t*	no	*t*=n.t.l.	Join words, alternating bold and italic.
`.BR`*t*	no	*t*=n.t.l.	Join words, alternating bold and roman.
`.DT`	no	.5i 1i...	Restore default tabs.
`.HP`*i*	yes	*i*=p.i.*	Begin paragraph with hanging indent. Set prevailing indent to *i*.
`.I`*t*	no	*t*=n.t.l.	Text is italic.
`.IB`*t*	no	*t*=n.t.l.	Join words, alternating italic and bold.
`.IP`*x i*	yes	*x*=""	Same as `.TP` with tag *x*.
`.IR`*t*	no	*t*=n.t.l.	Join words, alternating italic and roman.

Request	*Cause*	*If no*	*Explanation*
	Break	*Argument*	
.IX*t*	no	-	Index macro, for SunSoft internal use.
.LP	yes	-	Begin left-aligned paragraph. Set prevailing indent to .5i.
.P	yes	-	Same as .LP.
.PD*d*	no	*d*=.4v	Set vertical distance between paragraphs.
.PP	yes	-	Same as .LP.
.RE	yes	-	End of relative indent. Restores prevailing indent.
.RB*t*	no	*t*=n.t.l.	Join words, alternating roman and bold.
.RI*t*	no	*t*=n.t.l.	Join words, alternating roman and italic.
.RS*i*	yes	*i*=p.i.	Start relative indent, increase indent by *i*. Sets prevailing indent to .5i for nested indents.
.SB*t*	no	-	Reduce size of text by 1 point, make text bold.
.SH*t*	yes	-	Section Heading.
.SM*t*	no	*t*=n.t.l.	Reduce size of text by 1 point.
.SS*t*	yes	*t*=n.t.l.	Section Subheading.
.TH*n s d f m*	yes	-	Begin reference page *n*, of of section *s*; *d* is the date of the most recent change. If present, *f* is the left page footer; *m* is the main page (center) header. Sets prevailing indent and tabs to .5i.
.TP*i*	yes	*i*=p.i.	Begin indented paragraph, with the tag given on the next text line. Set prevailing indent to *i*.
.TX*t p*	no	-	Resolve the title abbreviation *t*; join to punctuation mark (or text) *p*.

Conventions

When formatting a manual page, man examines the first line to determine whether it requires special processing. For example a first line consisting of:

```
'\" t
```

indicates that the manual page must be run through the tbl(1) preprocessor.

A typical manual page for a command or function is laid out as follows:

.TH *title*[1-9]
The name of the command or function, which serves as the title of the manual page. This is followed by the number of the section in which it appears.

.SH NAME
The name, or list of names, by which the command is called, followed by a dash and then a one-line summary of the action performed. All in roman font, this section contains no `troff(1)` commands or escapes, and no macro requests.

.SH SYNOPSIS

Commands:
The syntax of the command and its arguments, as typed on the command line. When in boldface, a word must be typed exactly as printed. When in italics, a word can be replaced with an argument that you supply. References to bold or italicized items are not capitalized in other sections, even when they begin a sentence.

Syntactic symbols appear in roman face:

[ ]
An argument, when surrounded by brackets is optional.

|
Arguments separated by a vertical bar are exclusive. You can supply only one item from such a list.

. . .
Arguments followed by an ellipsis can be repeated. When an ellipsis follows a bracketed set, the expression within the brackets can be repeated.

Functions:
If required, the data declaration, or `#include` directive, is shown first, followed by the function declaration. Otherwise, the function declaration is shown.

.SH DESCRIPTION
A narrative overview of the command or function's external behavior. This includes how it interacts with files or data, and how it handles the standard input, standard output and standard error. Internals and implementation details are normally omitted. This section attempts to provide a succinct overview in answer to the question, *what does it do?*

Literal text from the synopsis appears in constant width, as do literal filenames and references to items that appear elsewhere in the reference manuals. Arguments are italicized.

If a command interprets either subcommands or an input grammar, its command interface or input grammar is normally described in a `USAGE` section, which follows the `OPTIONS` section. The `DESCRIPTION` section only describes the behavior of the command itself, not that of subcommands.

.SH OPTIONS
The list of options along with a description of how each affects the command's operation.

.SH RETURN VALUES
A list of the values the library routine returns to the calling program and the conditions that cause these values to be returned.

.SH EXIT STATUS
A list of the values the utility returns to the calling program or shell, and the conditions that cause these values to be returned.

.SH FILES
A list of files associated with the command or function.

.SH SEE ALSO
A comma-separated list of related manual pages, followed by references to other published materials.

.SH DIAGNOSTICS
A list of diagnostic messages and an explanation of each.

.SH BUGS
A description of limitations, known defects, and possible problems associated with the command or function.

SMF service `svc:/application/man-index` is used to generate the index files for `-f`, `-k`, and `-K` options of man(1). The SMF service is online by default. The index files for the options are generated during package/system installation if the FMRI is specified as an `restart_fmri` actuator. Any man page delivering packages that use the FMRI as `restart_fmri` actuator for its file actions must deliver a unique symbolic link to their man page directory at the following directory:

```
/usr/share/man/index.d/
```

The unique symbolic link names should be of package FMRI without scheme, slash character (`/`) replaced with `"%2F"`, and optionally have some or all sequences of numbers of version. For instance, for the following package FMRI:

```
pkg://opensolaris.org/library/libc@5.11,5.11-0.75:20071001T163427
```

could supply one of the symbolic link names such as:

```
opensolaris.org%2Flibrary%2Flibc
opensolaris.org%2Flibrary%2Flibc@5.11
```

The method of the SMF service, upon requested to generate index files using `restart_fmri` during package install, checks the symbolic links. Based on the newness of the installed files, index files are generated only as necessary.

Unless used in an IPS package as `restart_fmri`, by default, it generates index files in `/usr/share/man/` and `/usr/gnu/share/man/`. When used in and IPS package as `restart_fmri`, it generates and places index files in the same directory as the man page source directory of the package.

ファイル

`/usr/share/man/index.d/`	Directory to save symbolic links to man page directories needing index files for `-f`, `-k`, and `-K` queries.
`/usr/share/man/man_index/*`	Table of Contents and keyword database. Generated files include: ■ `/usr/share/man/man_index/man.idx` ■ `/usr/share/man/man_index/man.dic` ■ `/usr/share/man/man_index/man.frq` ■ `/usr/share/man/man_index/man.pos`
`/usr/share/lib/tmac/an`	

関連項目

`man(1)`, `nroff(1)`, `troff(1)`, `whatis(1)`

Dale Dougherty and Tim O'Reilly, Unix Text Processing

名前

mansun – macros to format Reference Manual pages

形式

`nroff -mansun` *filename*...

`troff -mansun` *filename*...

機能説明

These macros are used to lay out the reference pages in this manual. Note: if *filename* contains format input for a preprocessor, the commands shown above must be piped through the appropriate preprocessor. This is handled automatically by `man(1)`. See the "Conventions" section.

Any text argument *t* may be zero to six words. Quotes may be used to include SPACE characters in a "word". If *text* is empty, the special treatment is applied to the next input line with text to be printed. In this way `.I` may be used to italicize a whole line, or `.SB` may be used to make small bold letters.

A prevailing indent distance is remembered between successive indented paragraphs, and is reset to default value upon reaching a non-indented paragraph. Default units for indents *i* are ens.

Type font and size are reset to default values before each paragraph, and after processing font and size setting macros.

These strings are predefined by `-mansun`:

`\*R` '®', '(Reg)' in `nroff`.

`\*S` Change to default type size.

Requests

* n.t.l. = next text line; p.i. = prevailing indent

Request	*Cause Break*	*If no Argument*	*Explanation*
`.B` *t*	no	*t*=n.t.l.*	Text is in bold font.
`.BI` *t*	no	*t*=n.t.l.	Join words, alternating bold and italic.
`.BR` *t*	no	*t*=n.t.l.	Join words, alternating bold and Roman.
`.DT`	no	.5i 1i...	Restore default tabs.
`.HP` *i*	yes	*i*=p.i.*	Begin paragraph with hanging indent. Set prevailing indent to *i*.
`.I` *t*	no	*t*=n.t.l.	Text is italic.
`.IB` *t*	no	*t*=n.t.l.	Join words, alternating italic and bold.
`.IP` *x i*	yes	*x*=""	Same as `.TP` with tag *x*.

Request	*Cause Break*	*If no Argument*	*Explanation*
`.IR` *t*	no	*t*=n.t.l.	Join words, alternating italic and Roman.
`.IX` *t*	no	-	Index macro, for SunSoft internal use.
`.LP`	yes	-	Begin left-aligned paragraph. Set prevailing indent to .5i.
`.P`	yes	-	Same as `.LP`.
`.PD` *d*	no	*d*=.4v	Set vertical distance between paragraphs.
`.PP`	yes	-	Same as `.LP`.
`.RE`	yes	-	End of relative indent. Restores prevailing indent.
`.RB` *t*	no	*t*=n.t.l.	Join words, alternating Roman and bold.
`.RI` *t*	no	*t*=n.t.l.	Join words, alternating Roman and italic.
`.RS` *i*	yes	*i*=p.i.	Start relative indent, increase indent by *i*. Sets prevailing indent to .5i for nested indents.
`.SB` *t*	no	-	Reduce size of text by 1 point, make text bold.
`.SH` *t*	yes	-	Section Heading.
`.SM` *t*	no	*t*=n.t.l.	Reduce size of text by 1 point.
`.SS` *t*	yes	*t*=n.t.l.	Section Subheading.
`.TH` *n s d f m*	yes	-	Begin reference page *n*, of of section *s*; *d* is the date of the most recent change. If present, *f* is the left page footer; *m* is the main page (center) header. Sets prevailing indent and tabs to .5i.
`.TP` *i*	yes	*i*=p.i.	Begin indented paragraph, with the tag given on the next text line. Set prevailing indent to *i*.
`.TX` *t p*	no	-	Resolve the title abbreviation *t*; join to punctuation mark (or text) *p*.

Conventions

When formatting a manual page, `mansun` examines the first line to determine whether it requires special processing. For example a first line consisting of:

```
'\" t
```

indicates that the manual page must be run through the `tbl(1)` preprocessor.

A typical manual page for a command or function is laid out as follows:

.TH *title* [1-8] The name of the command or function, which serves as the title of the manual page. This is followed by the number of the section in which it appears.

.SH NAME The name, or list of names, by which the command is called, followed by a dash and then a one-line summary of the action performed. All in Roman font, this section contains no `troff(1)` commands or escapes, and no macro requests. It is used to generate the `windex` database, which is used by the `whatis(1)` command.

.SH SYNOPSIS

Commands: The syntax of the command and its arguments, as typed on the command line. When in boldface, a word must be typed exactly as printed. When in italics, a word can be replaced with an argument that you supply. References to bold or italicized items are not capitalized in other sections, even when they begin a sentence.

Syntactic symbols appear in Roman face:

[ ] An argument, when surrounded by brackets is optional.

| Arguments separated by a vertical bar are exclusive. You can supply only one item from such a list.

. . . Arguments followed by an ellipsis can be repeated. When an ellipsis follows a bracketed set, the expression within the brackets can be repeated.

Functions: If required, the data declaration, or `#include` directive, is shown first, followed by the function declaration. Otherwise, the function declaration is shown.

.SH DESCRIPTION A narrative overview of the command or function's external behavior. This includes how it interacts with files or data, and how it handles the standard input, standard output and standard error. Internals and implementation details are normally omitted. This section attempts to provide a succinct overview in answer to the question, "what does it do?"

Literal text from the synopsis appears in constant width, as do literal filenames and references to items that appear elsewhere in the reference manuals. Arguments are italicized.

If a command interprets either subcommands or an input grammar, its command interface or input grammar is normally described in a `USAGE`

	section, which follows the OPTIONS section. The DESCRIPTION section only describes the behavior of the command itself, not that of subcommands.
.SH OPTIONS	The list of options along with a description of how each affects the command's operation.
.SH FILES	A list of files associated with the command or function.
.SH SEE ALSO	A comma-separated list of related manual pages, followed by references to other published materials.
.SH DIAGNOSTICS	A list of diagnostic messages and an explanation of each.
.SH BUGS	A description of limitations, known defects, and possible problems associated with the command or function.

ファイル

`/usr/share/lib/tmac/ansun`

`/usr/share/man/windex`

関連項目

`man(1)`, `nroff(1)`, `troff(1)`, `whatis(1)`

Dale Dougherty and Tim O'Reilly, Unix Text Processing

名前

me – macros for formatting papers

形式

`nroff -me [`*options*`] `*filename*`...`

`troff -me [`*options*`] `*filename*`...`

機能説明

This package of `nroff` and `troff` macro definitions provides a canned formatting facility for technical papers in various formats. When producing 2-column output on a terminal, filter the output through `col(1)`.

The macro requests are defined below. Many `nroff` and `troff` requests are unsafe in conjunction with this package, however, these requests may be used with impunity after the first `.pp`:

.bp	begin new page
.br	break output line here
.sp *n*	insert *n* spacing lines
.ls *n*	(line spacing) *n*=1 single, *n*=2 double space
.na	no alignment of right margin
.ce *n*	center next *n* lines
.ul *n*	underline next *n* lines
.sz +*n*	add *n* to point size

Output of the `eqn(1)`, `neqn(1)`, `refer(1)`, and `tbl(1)` preprocessors for equations and tables is acceptable as input.

Requests

In the following list, "initialization" refers to the first `.pp`, `.lp`, `.ip`, `.np`, `.sh`, or `.uh` macro. This list is incomplete.

Request	*Initial Value*	*Cause Break*	*Explanation*
`.(c`	-	yes	Begin centered block.
`.(d`	-	no	Begin delayed text.
`.(f`	-	no	Begin footnote.
`.(l`	-	yes	Begin list.
`.(q`	-	yes	Begin major quote.
`.(x`*x*	-	no	Begin indexed item in index *x*.
`.(z`	-	no	Begin floating keep.

Request	*Initial Value*	*Cause Break*	*Explanation*
`.)c`	-	yes	End centered block.
`.)d`	-	yes	End delayed text.
`.)f`	-	yes	End footnote.
`.)l`	-	yes	End list.
`.)q`	-	yes	End major quote.
`.)x`	-	yes	End index item.
`.)z`	-	yes	End floating keep.
`.++` *m H*	-	no	Define paper section. *m* defines the part of the paper, and can be `C` (chapter), `A` (appendix), `P` (preliminary, for instance, abstract, table of contents, etc.), `B` (bibliography), `RC` (chapters renumbered from page one each chapter), or `RA` (appendix renumbered from page one).
`.+c` *T*	-	yes	Begin chapter (or appendix, etc., as set by `.++`). *T* is the chapter title.
`.1c`	1	yes	One column format on a new page.
`.2c`	1	yes	Two column format.
`.EN`	-	yes	Space after equation produced by `eqn` or `neqn`.
`.EQ` *x y*	-	yes	Precede equation; break out and add space. Equation number is *y*. The optional argument *x* may be *I* to indent equation (default), *L* to left-adjust the equation, or

Request	*Initial Value*	*Cause Break*	*Explanation*
			C to center the equation.
.GE	-	yes	End *gremlin* picture.
.GS	-	yes	Begin *gremlin* picture.
.PE	-	yes	End pic picture.
.PS	-	yes	Begin pic picture.
.TE	-	yes	End table.
.TH	-	yes	End heading section of table.
.TS *x*	-	yes	Begin table; if *x* is *H* table has repeated heading.
.ac *A N*	-	no	Set up for ACM style output. *A* is the Author's name(s), *N* is the total number of pages. Must be given before the first initialization.
.b *x*	no	no	Print *x* in boldface; if no argument switch to boldface.
.ba +*n*	0	yes	Augments the base indent by *n*. This indent is used to set the indent on regular text (like paragraphs).
.bc	no	yes	Begin new column.
.bi *x*	no	no	Print *x* in bold italics (nofill only).
.bu	-	yes	Begin bulleted paragraph.
.bx *x*	no	no	Print *x* in a box (nofill only).
.ef *'x'y'z*	"""	no	Set even footer to *x y z*.
.eh *'x'y'z*	"""	no	Set even header to *x y z*.
.fo *'x'y'z*	"""	no	Set footer to *x y z*.
.hx	-	no	Suppress headers and footers on

Request	*Initial Value*	*Cause Break*	*Explanation*
			next page.
`.he` *'x'y'z*	"""	no	Set header to *x y z*.
`.hl`	-	yes	Draw a horizontal line.
`.i` *x*	no	no	Italicize *x*; if *x* missing, italic text follows.
`.ip` *x y*	no	yes	Start indented paragraph, with hanging tag *x*. Indentation is *y* ens (default 5).
`.lp`	yes	yes	Start left-blocked paragraph.
`.lo`	-	no	Read in a file of local macros of the form `.*`*x*. Must be given before initialization.
`.np`	1	yes	Start numbered paragraph.
`.of` *'x'y'z*	"""	no	Set odd footer to x y z.
`.oh` *'x'y'z*	"""	no	Set odd header to x y z.
`.pd`	-	yes	Print delayed text.
`.pp`	no	yes	Begin paragraph. First line indented.
`.r`	yes	no	Roman text follows.
`.re`	-	no	Reset tabs to default values.
`.sc`	no	no	Read in a file of special characters and diacritical marks. Must be given before initialization.
`.sh` *n x*	-	yes	Section head follows, font automatically bold. *n* is level of section, *x* is title of section.
`.sk`	no	no	Leave the next page blank. Only one page is remembered ahead.

Request	*Initial Value*	*Cause Break*	*Explanation*
`.sm` *x*	-	no	Set *x* in a smaller pointsize.
`.sz` +*n*	10p	no	Augment the point size by *n* points.
`.th`	no	no	Produce the paper in thesis format. Must be given before initialization.
`.tp`	no	yes	Begin title page.
`.u` *x*	-	no	Underline argument (even in `troff`). (Nofill only).
`.uh`	-	yes	Like `.sh` but unnumbered.
`.xp` *x*	-	no	Print index *x*.

ファイル

`/usr/share/lib/tmac/e`

`/usr/share/lib/tmac/*.me`

関連項目

`col`(1), `eqn`(1), `nroff`(1), `refer`(1), `tbl`(1), `troff`(1)

名前

mech_spnego – Simple and Protected GSS-API Negotiation Mechanism

形式

`/usr/lib/gss/mech_spnego.so.1`

機能説明

The SPNEGO security mechanism for GSS-API allows GSS-API applications to negotiate the actual security mechanism to be used in the GSS-API session. `mech_spnego.so.1` is a shared object module that is dynamically opened by applications that specify the SPNEGO Object Identifier (OID) in calls to the GSS-API functions (see `libgss(3LIB)`).

SPNEGO is described by IETF RFC 2478 and is intended to be used in environments where multiple GSS-API mechanisms are available to the client or server and neither side knows what mechanisms are supported by the other.

When SPNEGO is used, it selects the list of mechanisms to advertise by reading the GSS mechanism configuration file, `/etc/gss/mech` (see `mech(4)`), and by listing all active mechanisms except for itself.

オプション

SPNEGO may be configured to function in two ways. The first way is to interoperate with Microsoft SSPI clients and servers that use the Microsoft "`Negotiate`" method, which is also based on SPNEGO. The Microsoft "`Negotiate`" mechanism does not strictly follow the IETF RFC. Therefore, use special handling in order to enable full interoperability. In order to interoperate, place option "`[ msinterop ]`" at the end of the SPNEGO line in `/etc/gss/mech`.

This is an example (from `/etc/gss/mech`):

```
spnego    1.3.6.1.5.5.2    mech_spnego.so    [ msinterop ]
```

Without the "`[ msinterop ]`" option, `mech_spnego` will follow the strict IETF RFC 2478 specification and will not be able to negotiate with Microsoft applications that try to use the SSPI "`Negotiate`" mechanism.

Interfaces

`mech_spnego.so.1` has no public interfaces. It is only activated and used through the GSS-API interface provided by `libgss.so.1` (see `libgss(3LIB)`).

ファイル

`/usr/lib/gss/mech_spnego.so.1`	shared object file
`/usr/lib/sparcv9/gss/mech_spnego.so.1`	SPARC 64-bit shared object file
`/usr/lib/amd64/gss/mech_spnego.so.1`	x86 64-bit shared object file

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUWNspnego
MT Level	Safe

関連項目 Intro(3), libgss(3LIB), mech(4), attributes(5)

『Oracle Solaris 11 セキュリティーサービス開発ガイド』

名前

mm – text formatting (memorandum) macros

形式

`nroff -mm [`*options*`] `*filename*`...`

`troff -mm [`*options*`] `*filename*`...`

機能説明

This package of `nroff(1)` and `troff(1)` macro definitions provides a formatting facility for various styles of articles, theses, and books. When producing 2-column output on a terminal or lineprinter, or when reverse line motions are needed, filter the output through `col(1)`. All external `-mm` macros are defined below.

Note: this `-mm` macro package is an extended version written at Berkeley and is a superset of the standard `-mm` macro packages as supplied by Bell Labs. Some of the Bell Labs macros have been removed; for instance, it is assumed that the user has little interest in producing headers stating that the memo was generated at Whippany Labs.

Many `nroff` and `troff` requests are unsafe in conjunction with this package. However, the first four requests below may be used with impunity after initialization, and the last two may be used even before initialization:

`.bp` begin new page

`.br` break output line

`.sp`*n* insert n spacing lines

`.ce`*n* center next n lines

`.ls`*n* line spacing: *n*=1 single, *n*=2 double space

`.na` no alignment of right margin

Font and point size changes with `\f` and `\s` are also allowed; for example, `\fIword\fR` will italicize *word*. Output of the `tbl(1)`, `eqn(1)` and `refer(1)` preprocessors for equations, tables, and references is acceptable as input.

Requests

Here is a table of macros.

Macro Name	Initial Value	Break? Reset?	Explanation
`.1C`	on	y,y	one column format on a new page
`.2C` [*l*]	–	y,y	two column format *l*=line length
`.AE`	–	y	end abstract
`.AL` [*t*] [*i*] [*s*]	*t*=1;*i*=`.Li`;*s*=`0`	y	Start automatic list type *t*=[`1`,`A`,`a`,`I`,`i`] `1`=arabic numbers; `A`=uppercase letters `a`=lowercase letters; `I`=uppercase Roman numerals; `i`=lowercase Roman numerals indentation *i*; separation *s*

Macro Name	Initial Value	Break? Reset?	Explanation
.AS *m* [*n*]	*n*=0	y	begin abstract
.AU	–	y	author's name
.AV *x*	–	y	signature and date line of verifier *x*
.B *x*	–	n	embolden *x*; if no *x*, switch to boldface
.BE	–	y	end block text
.BI *x y*	–	n	embolden *x* and underline *y*
.BL	–	y	bullet list
.BR *x y*	–	n	embolden *x* and use Roman font for *y*
.BS	–	n	start block text
.CN	–	y	same as .DE (nroff)
.CS	–	y	cover sheet
.CW	–	n	same as .DS I (nroff)
.DE	–	y	end display
.DF [*p*] [*f*] [*rp*]	*p*=L;*f*=N	y	start floating display; position *p*=[L,C,CB] L=left; I=indent; C=center; CB=center block fill *f*=[N,Y]; right position *rp* (fill only)
.DL [*i*] [*s*]	–	y	start dash list
.DS [*p*] [*f*] [*rp*]	*p*=L;*f*=N	y	begin static display (see .DF for argument descriptions)
.EC *x* [*n*]	*n*=1	y	equation title; equation *x*; number *n*
.EF *x*	–	n	even footer appears at the bottom of even-numbered pages; *x*="*l* ' *c* ' *r*" *l*=left; *c*=center; *r*=right
.EH *x*	–	n	even header appears at the top of even-numbered pages; *x*="*l* ' *c* ' *r*" *l*=left; *c*=center; *r*=right
.EN	–	y	end displayed equation produced by eqn
.EQ	–	y	break out equation produced by eqn
.EX *x* [*n*]	*n*=1	y	exhibit title; exhibit *x*
			number *n*

Macro Name	Initial Value	Break? Reset?	Explanation
.FD [*f*] [*r*]	*f*=0;*r*=1	n	set footnote style format *f*=[0-11]; renumber *r*=[0,1]
.FE	–	y	end footnote
.FG *x* [*n*]	*n*=1	y	figure title; figure *x*; number *n*
.FS	–	n	start footnote
.H *l* [*t*]	–	y	produce numbered heading level *l*=[1-7]; title *t*
.HU *t*	–	y	produce unnumbered heading; title *t*
.I *x*	–	n	underline *x*
.IB *x y*	–	n	underline *x* and embolden *y*
.IR *x y*	–	n	underline *x* and use Roman font on *y*
.LE [*s*]	*s*=0	y	end list; separation *s*
.LI [*m*] [*p*]	–	y	start new list item; mark *m*
			prefix *p* (mark only)
.ML *m* [*i*] [*s*]	*s*=0	y	start marked list; mark *m* indentation *i*; separation *s*=[0,1]
.MT *x*		y	memo title; title *x*
.ND *x*		n	no date in page footer; *x* is date on cover
.NE	–	y	end block text
.NS	–	y	start block text
.OF *x*	–	n	odd footer appears at the bottom of odd-numbered pages; *x*="*l* ' *c* ' *r*" *l*=left; *c*=center; *r*=right
.OF *x*	–	n	odd header appears at the top of odd-numbered pages; *x*="*l* ' *c* ' *r*" *l*=left; *c*=center; *r*=right
.OP	–	y	skip to the top of an odd-number page
.P [*t*]	*t*=0	y,y	begin paragraph; *t*=[0,1] 0=justified; 1=indented
.PF *x*	–	n	page footer appears at the bottom of every page; *x*="*l* ' *c* ' *r*" *l*=left; *c*=center; *r*=right

Macro Name	Initial Value	Break? Reset?	Explanation
.PH *x*	–	n	page header appears at the top of every page; *x*="*l' c' r*" *l*=left; *c*=center; *r*=right
.R	on	n	return to Roman font
.RB *x y*	–	n	use Roman on *x* and embolden *y*
.RI *x y*	–	n	use Roman on *x* and underline *y*
.RP *x*	-	y,y	released paper format ? *x*=no stops title on first
.RS	5n	y,y	right shift: start level of relative indentation
.S *m n*	–	n	set character point size & vertical space character point size *m*; vertical space *n*
.SA *x*	*x*=1	n	justification; *x*=[0,1]
.SK *x*	–	y	skip *x* pages
.SM	–	n	smaller; decrease point size by 2
.SP [*x*]	–	y	leave *x* blank lines
.TB *x* [*n*]	*n*=1	y	table title; table *x*; number *n*
.TC	–	y	print table of contents (put at end of input file)
.TE	–	y	end of table processed by tbl
.TH	–	y	end multi-page header of table
.TL	–	n	title in boldface and two points larger
.TM	–	n	UC Berkeley thesis mode
.TP *i*	y	y	*i*=p.i. Begin indented paragraph, with the tag given on the next text line. Set prevailing indent to *i*.
.TS *x*	–	y,y	begin table; if *x*=H table has multi-page header
.TY	–	y	display centered title CONTENTS
.VL *i* [*m*] [*s*]	*m*=0;*s*=0	y	start variable-item list; indentation *i* mark-indentation *m*; separation *s*

Registers

Formatting distances can be controlled in -mm by means of built-in number registers. For example, this sets the line length to 6.5 inches:

```
.nr  LL  6.5i
```

Here is a table of number registers and their default values:

Name	Register Controls	Takes Effect	Default
Cl	contents level	table of contents	2
De	display eject	display	0
Df	display floating	display	5
Ds	display spacing	display	1v
Hb	heading break	heading	2
Hc	heading centering	heading	0
Hi	heading indent	heading	1
Hi	heading spacing	heading	1
Hu	heading unnumbered	heading	2
Li	list indentation	list	6 (nroff) 5 (troff)
Ls	list spacing	list	6
Pi	paragraph indent	paragraph	5
Pt	paragraph type	paragraph	1
Si	static indent	display	5 (nroff) 3 (troff)

When resetting these values, make sure to specify the appropriate units. Setting the line length to 7, for example, will result in output with one character per line. Setting Pi to 0 suppresses paragraph indentation

Here is a list of string registers available in -mm; they may be used anywhere in the text:

Name	String's Function
*Q	quote (" in nroff, “ in troff)
*U	unquote (" in nroff, ’’ in troff)
*–	dash (-- in nroff, — in troff)
*(MO	month (month of the year)

Name	String's Function
*(DY	day (current date)
**	automatically numbered footnote
*'	acute accent (before letter)
*`	grave accent (before letter)
*^	circumflex (before letter)
*,	cedilla (before letter)
*:	umlaut (before letter)
*~	tilde (before letter)
\(BU	bullet item
\(DT	date (*month day*, *yr*)
\(EM	em dash
\(Lf	LIST OF FIGURES title
\(Lt	LIST OF TABLES title
\(Lx	LIST OF EXHIBITS title
\(Le	LIST OF EQUATIONS title
\(Rp	REFERENCES title
\(Tm	trademark character (TM)

When using the extended accent mark definitions available with .AM, these strings should come after, rather than before, the letter to be accented.

ファイル

/usr/share/lib/tmac/m

/usr/share/lib/tmac/mm.[nt] nroff and troff definitions of mm.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

関連項目

col(1), eqn(1), nroff(1), refer(1), tbl(1), troff(1), attributes(5)

使用上の留意点 Floating keeps and regular keeps are diverted to the same space, so they cannot be mixed together with predictable results.

名前

ms – text formatting macros

形式

```
nroff -ms [options] filename...
troff -ms [options] filename...
```

機能説明

This package of nroff(1) and troff(1) macro definitions provides a formatting facility for various styles of articles, theses, and books. When producing 2-column output on a terminal or lineprinter, or when reverse line motions are needed, filter the output through col(1). All external -ms macros are defined below.

Note: this -ms macro package is an extended version written at Berkeley and is a superset of the standard -ms macro packages as supplied by Bell Labs. Some of the Bell Labs macros have been removed; for instance, it is assumed that the user has little interest in producing headers stating that the memo was generated at Whippany Labs.

Many nroff and troff requests are unsafe in conjunction with this package. However, the first four requests below may be used with impunity after initialization, and the last two may be used even before initialization:

.bp	begin new page
.br	break output line
.sp *n*	insert n spacing lines
.ce *n*	center next n lines
.ls *n*	line spacing: *n*=1 single, *n*=2 double space
.na	no alignment of right margin

Font and point size changes with \f and \s are also allowed; for example, \fIword\fR will italicize *word*. Output of the tbl(1), eqn(1) and refer(1) preprocessors for equations, tables, and references is acceptable as input.

Requests

Macro Name	Initial Value	Break? Reset?	Explanation
.AB *x*	–	y	begin abstract; if *x*=no do not label abstract
.AE	–	y	end abstract
.AI	–	y	author's institution
.AM	–	n	better accent mark definitions
.AU	–	y	author's name
.B *x*	–	n	embolden *x*; if no *x*, switch to boldface
.B1	–	y	begin text to be enclosed in a box

Macro Name	Initial Value	Break? Reset?	Explanation
.B2	–	y	end boxed text and print it
.BT	date	n	bottom title, printed at foot of page
.BX *x*	–	n	print word *x* in a box
.CM	if t	n	cut mark between pages
.CT	–	y,y	chapter title: page number moved to CF (TM only)
.DA *x*	if n	n	force date *x* at bottom of page; today if no *x*
.DE	–	y	end display (unfilled text) of any kind
.DS *x y*	I	y	begin display with keep; *x*=I, L, C, B; *y*=indent
.ID *y*	8n,.5i	y	indented display with no keep; *y*=indent
.LD	–	y	left display with no keep
.CD	–	y	centered display with no keep
.BD	–	y	block display; center entire block
.EF *x*	–	n	even page footer *x* (3 part as for .tl)
.EH *x*	–	n	even page header *x* (3 part as for .tl)
.EN	–	y	end displayed equation produced by eqn
.EQ *x y*	–	y	break out equation; *x*=L,I,C; *y*=equation number
.FE	–	n	end footnote to be placed at bottom of page
.FP	–	n	numbered footnote paragraph; may be redefined
.FS *x*	–	n	start footnote; *x* is optional footnote label
.HD	undef	n	optional page header below header margin
.I *x*	–	n	italicize *x*; if no *x*, switch to italics
.IP *x y*	–	y,y	indented paragraph, with hanging tag *x*; *y*=indent
.IX *x y*	–	y	index words *x y* and so on (up to 5 levels)
.KE	–	n	end keep of any kind
.KF	–	n	begin floating keep; text fills remainder of page
.KS	–	y	begin keep; unit kept together on a single page
.LG	–	n	larger; increase point size by 2
.LP	–	y,y	left (block) paragraph.

Macro Name	Initial Value	Break? Reset?	Explanation
.MC *x*	–	y,y	multiple columns; *x*=column width
.ND *x*	if t	n	no date in page footer; *x* is date on cover
.NH *x y*	–	y,y	numbered header; *x*=level, *x*=0 resets, *x*=S sets to *y*
.NL	10p	n	set point size back to normal
.OF *x*	–	n	odd page footer *x* (3 part as for .tl)
.OH *x*	–	n	odd page header *x* (3 part as for .tl)
.P1	if TM	n	print header on first page
.PP	–	y,y	paragraph with first line indented
.PT	- % -	n	page title, printed at head of page
.PX *x*	–	y	print index (table of contents); *x*=no suppresses title
.QP	–	y,y	quote paragraph (indented and shorter)
.R	on	n	return to Roman font
.RE	5n	y,y	retreat: end level of relative indentation
.RP *x*	–	n	released paper format; *x*=no stops title on first page
.RS	5n	y,y	right shift: start level of relative indentation
.SH	–	y,y	section header, in boldface
.SM	–	n	smaller; decrease point size by 2
.TA	8n,5n	n	set TAB characters to 8n 16n . . . (nroff) or 5n 10n . . . (troff)
.TC *x*	–	y	print table of contents at end; *x*=no suppresses title
.TE	–	y	end of table processed by tbl
.TH	–	y	end multi-page header of table
.TL	–	y	title in boldface and two points larger
.TM	off	n	UC Berkeley thesis mode
.TS *x*	–	y,y	begin table; if *x*=H table has multi-page header
.UL *x*	–	n	underline *x*, even in troff
.UX *x*	–	n	UNIX; trademark message first time; *x* appended
.XA *x y*	–	y	another index entry; *x*=page or no for none; y=indent

Macro Name	Initial Value	Break? Reset?	Explanation
.XE	–	y	end index entry (or series of .IX entries)
.XP	–	y,y	paragraph with first line indented, others indented
.XS *x y*	–	y	begin index entry; *x*=page or no for none; *y*=indent
.1C	on	y,y	one column format, on a new page
.2C	–	y,y	begin two column format
.] –	–	n	beginning of refer reference
.[0	–	n	end of unclassifiable type of reference
.[N	–	n	N= 1:journal-article, 2:book, 3:book-article, 4:report

Registers

Formatting distances can be controlled in -ms by means of built-in number registers. For example, this sets the line length to 6.5 inches:

```
.nr  LL  6.5i
```

Here is a table of number registers and their default values:

Name	Register Controls	Takes Effect	Default
PS	point size	paragraph	10
VS	vertical spacing	paragraph	12
LL	line length	paragraph	6i
LT	title length	next page	same as LL
FL	footnote length	next .FS	5.5i
PD	paragraph distance	paragraph	1v (if n), .3v (if t)
DD	display distance	displays	1v (if n), .5v (if t)
PI	paragraph indent	paragraph	5n
QI	quote indent	next .QP	5n
FI	footnote indent	next .FS	2n
PO	page offset	next page	0 (if n), ≈1i (if t)
HM	header margin	next page	1i
FM	footer margin	next page	1i
FF	footnote format	next .FS	0 (1, 2, 3 available)

When resetting these values, make sure to specify the appropriate units. Setting the line length to 7, for example, will result in output with one character per line. Setting FF to 1 suppresses footnote superscripting; setting it to 2 also suppresses indentation of the first line; and setting it to 3 produces an .IP-like footnote paragraph.

Here is a list of string registers available in -ms; they may be used anywhere in the text:

Name	String's Function
*Q	quote (" in nroff, “ in troff)
*U	unquote (" in nroff, ” in troff)
*–	dash (-- in nroff, – in troff)
*(MO	month (month of the year)
*(DY	day (current date)
**	automatically numbered footnote
*’	acute accent (before letter)
*‘	grave accent (before letter)
*^	circumflex (before letter)
*,	cedilla (before letter)
*:	umlaut (before letter)
*~	tilde (before letter)

When using the extended accent mark definitions available with .AM, these strings should come after, rather than before, the letter to be accented.

ファイル

/usr/share/lib/tmac/s

/usr/share/lib/tmac/ms.???

関連項目

col(1), eqn(1), nroff(1), refer(1), tbl(1), troff(1)

使用上の留意点

Floating keeps and regular keeps are diverted to the same space, so they cannot be mixed together with predictable results.

名前

mutex – concepts relating to mutual exclusion locks

機能説明

Mutual exclusion locks (mutexes) prevent multiple threads from simultaneously executing critical sections of code which access shared data (that is, mutexes are used to serialize the execution of threads). All mutexes must be global. A successful call to acquire a mutex will cause another thread that is also trying to lock the same mutex to block until the owner thread unlocks the mutex.

Mutexes can synchronize threads within the same process or in other processes. Mutexes can be used to synchronize threads between processes if the mutexes are allocated in writable memory and shared among the cooperating processes (see `mmap(2)`), and have been initialized for this task.

The following table lists mutex functions and the actions they perform.

FUNCTION	ACTION
`mutex_init`	Initialize a mutex.
`mutex_destroy`	Destroy a mutex.
`mutex_lock`	Lock a mutex.
`mutex_trylock`	Attempt to lock a mutex.
`mutex_unlock`	Unlock a mutex.
`pthread_mutex_init`	Initialize a mutex.
`pthread_mutex_destroy`	Destroy a mutex.
`pthread_mutex_lock`	Lock a mutex.
`pthread_mutex_trylock`	Attempt to lock a mutex.
`pthread_mutex_unlock`	Unlock a mutex.

Initialization

Mutexes are either intra-process or inter-process, depending upon the argument passed implicitly or explicitly to the initialization of that mutex. A statically allocated mutex does not need to be explicitly initialized; by default, a statically allocated mutex is initialized with all zeros and its scope is set to be within the calling process.

For inter-process synchronization, a mutex needs to be allocated in memory shared between these processes. Since the memory for such a mutex must be allocated dynamically, the mutex needs to be explicitly initialized with the appropriate attribute that indicates inter-process use.

Locking and Unlocking

A critical section of code is enclosed by a call to lock the mutex and the call to unlock the mutex to protect it from simultaneous access by multiple threads. Only one thread at a time may possess mutually exclusive access to the critical section of code that is enclosed by the mutex-locking call and the mutex-unlocking call, whether the mutex's scope is intra-process

or inter-process. A thread calling to lock the mutex either gets exclusive access to the code starting from the successful locking until its call to unlock the mutex, or it waits until the mutex is unlocked by the thread that locked it.

Mutexes have ownership, unlike semaphores. Only the thread that locked a mutex, (that is, the owner of the mutex), should unlock it.

If a thread waiting for a mutex receives a signal, upon return from the signal handler, the thread resumes waiting for the mutex as if there was no interrupt.

Caveats

Mutexes are almost like data – they can be embedded in data structures, files, dynamic or static memory, and so forth. Hence, they are easy to introduce into a program. However, too many mutexes can degrade performance and scalability of the application. Because too few mutexes can hinder the concurrency of the application, they should be introduced with care. Also, incorrect usage (such as recursive calls, or violation of locking order, and so forth) can lead to deadlocks, or worse, data inconsistencies.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe

関連項目

mmap(2), shmop(2), mutex_destroy(3C), mutex_init(3C), mutex_lock(3C), mutex_trylock(3C), mutex_unlock(3C), pthread_create(3C), pthread_mutex_destroy(3C), pthread_mutex_init(3C), pthread_mutex_lock(3C), pthread_mutex_trylock(3C), pthread_mutex_unlock(3C), pthread_mutexattr_init(3C), attributes(5), standards(5)

注意事項

In the current implementation of threads, pthread_mutex_lock(), pthread_mutex_unlock(), mutex_lock() mutex_unlock(), pthread_mutex_trylock(), and mutex_trylock() do not validate the mutex type. Therefore, an uninitialized mutex or a mutex with an invalid type does not return EINVAL. Interfaces for mutexes with an invalid type have unspecified behavior.

By default, if multiple threads are waiting for a mutex, the order of acquisition is undefined.

The system does not support multiple mappings to the same logical synch object if it is initialized as process-private (USYNC_THREAD for Solaris, PTHREAD_PROCESS_PRIVATE for POSIX). If you need to mmap(2)a synch object to different locations within the same address space, then the synch object should be initialized as a shared object (USYNC_PROCESS for Solaris, PTHREAD_PROCESS_SHARED for POSIX).

名前

mwac, MWAC – Mandatory Write Access Control

機能説明

Mandatory Write Access Control (MWAC) implements a new policy in the Oracle Solaris operating environment, that allows for fine- grained control over the writability of objects on otherwise read-only file systems.

In the current instance of the Oracle Solaris operating environment, MWAC is available only to non-global zones. The global zone implements the MWAC policy for non-global zones, preventing any overruling of the policy from within the non-global zone.

Zones marked as `read-only` have their root file system write-protected by MWAC. Only the file system objects that are write-listed by the read-only-profile are writable. See `zonecfg`(1M). Other file system objects are read-only.

Creating links to objects that are read-only by virtue of the MWAC-policy is not allowed.

関連項目

`ln`(1), `zoneadm`(1M), `zonecfg`(1M), `link`(2), `pathconf`(2)

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

名前

nfssec – overview of NFS security modes

機能説明

The mount_nfs(1M) and share_nfs(1M) commands each provide a way to specify the security mode to be used on an NFS file system through the sec=*mode* option. *mode* can be sys, dh, krb5, krb5i, krb5p, or none. These security modes can also be added to the automount maps. mount_nfs(1M) allows you to specify a single security mode; share_nfs(1M) allows you to specify multiple modes (or none). With multiple modes, an NFS client can choose any of the modes in the list.

The sec=*mode* option on the share_nfs(1M) command line establishes the security mode of NFS servers. If the NFS connection uses the NFS Version 3 protocol, the NFS clients must query the server for the appropriate *mode* to use. If the NFS connection uses the NFS Version 2 protocol, then the NFS client uses the default security mode, which is currently sys. NFS clients may force the use of a specific security mode by specifying the sec=*mode* option on the command line. However, if the file system on the server is not shared with that security mode, the client may be denied access.

If the NFS client wants to authenticate the NFS server using a particular (stronger) security mode, the client wants to specify the security mode to be used, even if the connection uses the NFS Version 3 protocol. This guarantees that an attacker masquerading as the server does not compromise the client.

The NFS security modes are described below. Of these, the krb5, krb5i, krb5p modes use the Kerberos V5 protocol for authenticating and protecting the shared filesystems. Before these can be used, the system must be configured to be part of a Kerberos realm. See kerberos(5).

sys	Use AUTH_SYS authentication. The user's UNIX user-id and group-ids are passed in the clear on the network, unauthenticated by the NFS server. This is the simplest security method and requires no additional administration. It is the default used by Solaris NFS Version 2 clients and Solaris NFS servers.
dh	Use a Diffie-Hellman public key system (AUTH_DES, which is referred to as AUTH_DH in RFC 2695: Authentication Mechanisms for ONC RPC.
krb5	Use Kerberos V5 protocol to authenticate users before granting access to the shared filesystem.
krb5i	Use Kerberos V5 authentication with integrity checking (checksums) to verify that the data has not been tampered with.
krb5p	User Kerberos V5 authentication, integrity checksums, and privacy protection (encryption) on the shared filesystem. This provides the most secure filesystem sharing, as all traffic is encrypted. It should be noted that performance might suffer on some systems when using krb5p, depending on the computational intensity of the encryption algorithm and the amount of data being transferred.

`none`
: Use null authentication (`AUTH_NONE`). NFS clients using `AUTH_NONE` have no identity and are mapped to the anonymous user `nobody` by NFS servers. A client using a security mode other than the one with which a Solaris NFS server shares the file system has its security mode mapped to `AUTH_NONE`. In this case, if the file system is shared with `sec=`*none,* users from the client are mapped to the anonymous user. The NFS security mode `none` is supported by `share_nfs`(1M).

`sec=`*mode*[`:`*mode*]. . .
: Sharing uses one or more of the specified security modes. The *mode* in the `sec=`*mode* option must be a node name supported on the client. If the `sec=` option is not specified, the default security mode used is `AUTH_SYS`. Multiple `sec=` options can be specified on the command line, although each mode can appear only once.

  Each `sec=` option specifies modes that apply to any subsequent `window=, rw, ro, rw=, ro=` and `root=` options that are provided before another `sec=`option. Each additional `sec=` resets the security mode context, so that more `window=, rw, ro, rw=, ro=` and `root=` options can be supplied for additional modes.

The NFSv4 server constructs a shared file system name space which is identical to the real file system name space on the server, including directories which are not actually shared, if they lead to shared directories. The constructed parts of the name space are known as the `pseudo-fs`. The `pseudo-fs` is always read-only.

As with NFSv3, the security mode of the shared directory is controlled using the `sec=`*mode* option of `share_nfs`(1M). However, the security mode of `pseudo-fs` objects is the union of the various security modes of the shared directories below.

When an NFSv4 client performs a mount, the client traverses the server's name space, from the root, down to the directory being mounted. Using the features of the NFSv4 protocol, the client may negotiate the security flavor of the directories as it proceeds down. If no `sec=`*mode* option is given to `mount_nfs` or an automounter map entry, then the client will do full negotiation for each directory down to the mount point, changing security flavors as needed. If `sec=`*mode* option is given, the client is constrained to use the requested security mode for all operations.

使用例

例 1 Sharing `/var` with Kerberos Authentication and Integrity Protection

The following example shares `/var` with Kerberos authentication and integrity protection:

```
share -F nfs -o sec=krb5i /var
```

例 2 Sharing /var with Kerberos Authentication and Privacy Protection

The following example shares/var with Kerberos authentication and privacy protection:

```
share -F nfs -o sec=krb5p /var
```

例 3 Sharing /var with Kerberos Authentication and Optionally Falling Back to AUTH_SYS Authentication

The following example shares /var with Kerberos authentication and optionally falls back to AUTH_SYS authentication:

```
share -F nfs -o sec=krb5:sys /var
```

例 4 Sharing /var with Kerberos Authentication Allowing read/write Operations for Kerberos Authenticated Users and Optionally Falling Back to AUTH_SYS Authentication Allowing only Read Operations

The following example shares /var with Kerberos authentication allowing read/write operations for Kerberos authenticated users and optionally falls back to AUTH_SYS authentication allowing only read operations:

```
share -F nfs -o sec=krb5,rw,sec=sys,ro /var
```

ファイル

/etc/nfssec.conf NFS security service configuration file

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

関連項目

automount(1M), kclient(1M), mount_nfs(1M), share_nfs(1M), rpc_clnt_auth(3NSL), secure_rpc(3NSL), nfssec.conf(4), attributes(5), kerberos(5)

RFC 2695: Authentication Mechanisms for ONC RPC

注意事項

/etc/nfssec.conf lists the NFS security services. Do not edit this file. It is not intended to be user-configurable. See kclient(1M).

名前

nis, NIS, NIS+ – network information name service

機能説明

NIS, formerly known as the Yellow Pages or YP, is the name of the network information name service in common use in networks on which Sun and other vendors' network nodes reside. The most recent version of NIS is version 2.

All commands and functions that use NIS version 2 are prefixed by the letters yp as in `ypmatch(1)`, `ypcat(1)`, `yp_match(3NSL)`, and `yp_first(3NSL)`.

The NIS+ name service is no longer shipped with the Solaris operating system. Tools to aid the migration from NIS+ to LDAP are available in the current Solaris release.

名前

nwam – network auto-magic 構成の管理

機能説明

ネットワーク構成は、システムによって定義されたデフォルトポリシー規則またはユーザー定義のポリシー規則に従って、自動的に管理できます。この管理は nwamd デーモンによって実行されます。

NWAM 機能は network/physical:default SMF サービスによって提供されます。netcfg/active_ncp プロパティーがリアクティブな NCP の名前に設定されている場合は、NWAM 構成管理が有効になっています。一方、netcfg/active_ncp プロパティーが DefaultFixed に設定されている場合は従来のネットワーク構成が実行されます。

動作

自動 NCP で実装されているデフォルトの構成ポリシーでは、物理的に接続されている Ethernet リンクがすべてアクティブになり、使用可能な Ethernet リンクがない場合は 1 つの無線リンクがアクティブになります。アクティブなリンクの IP アドレスはすべて DHCP を使用して取得されます。このデフォルトポリシーは、代替のネットワーク構成プロファイル (NCP) を作成することによって変更できます。NCP を作成および変更するには、nwam-manager GUI ツールまたは nwamd(1M) コマンドを使用できます。

NWAM 管理はまた、ネームサービスやネットワークセキュリティープロトコルなどの上位レイヤーのネットワーク構成にも使用できます。この構成は、場所プロファイルで指定されます。

プロファイルは、netcfg(1M) を使用して管理できます。これは、すべてのリアクティブなプロファイルのアクティブ化ポリシーを指定するための主要なツールです。これはまた、場所や単純な NCP コンポーネントを指定するためにも使用できます。

dladm(1M) または ipadm(1M) を使用して行われた構成の変更は、その NCP がリアクティブ (つまり、nwamd によってアクティブに管理されている) か、または解決されたかにかかわらず、現在アクティブな NCP に適用されます。

場所は、より多様な一連の構成オブジェクトを管理します。これらのオブジェクトのアクティブなシステム構成は SMF 内に存在します。NWAM 機能は、設定をプロファイルリポジトリ内に格納し、場所がアクティブになったらそれらの設定を SMF リポジトリに適用することによってその構成を管理します。

リアクティブな場所の場合、場所がアクティブな間に SMF リポジトリに加えられた変更はすべて、その場所が無効になると失われます。代わりに、NWAM プロファイルへの変更は netcfg を使用して行うようにしてください。これらの変更は NWAM リポジトリに保存され、現在アクティブな場所に対して行われた場合は、アクティブなシステム構成にも適用されます。

DefaultFixed の場所の場合、SMF リポジトリに加えられた変更は、その場所が無効になると NWAM リポジトリに保存されます。そのため、次回 DefaultFixed の場所が有効になると復元されます。

サービスのプロパティー

形式は次のようになります。

property_group/property_name property_type default_value
説明

`nwamd/debug` boolean `false`
`daemon.debug` を使用したデバッグログを有効にします。

`nwamd/autoconf` boolean `false`
より適切な (より優先順位の高い) 選択肢がない場合に、オープンな WLAN に自動的に接続するかどうかを示します。

`nwamd/ncu_wait_time` count `60`
NCU (場合によっては、リンクとインタフェースの NCU ペア) が起動するのを待つ秒数。この時間が経過すると、次の使用可能な NCU を試します。起動操作は取り消されないため、最終的に成功する場合があります。その時点で、指定されている構成の条件に応じて、より優先順位の高い NCU がアクティブになり、他方は無効になることがあります。

`nwamd/condition_check_interval` count `120`
条件に応じてアクティブになるオブジェクトの条件確認を定期的に行う間隔 (秒単位)。最小値は 30 秒です。

`nwamd/scan_interval` count `120`
無線スキャンを定期的に行う間隔 (秒単位)。

`nwamd/scan_level` astring `weak`
信号強度のしきい値。現在接続している AP がこの信号レベルより低下した場合、より信号の強い同等の AP (同じ ESSID を持つ) が使用可能であれば、現在の接続は切断され、より信号の強い AP に接続されます。

`nwamd/strict_bssid` boolean `false`
true の場合、以前接続した WLAN に接続するには、ESSID と BSSID が一致する必要があります。false の場合、一致する必要があるのは ESSID だけです。

`netcfg/active_ncp` astring `Automatic`
現在アクティブな NCP。ユーザーはこのプロパティーを設定しないようにしてください。再起動後も持続されるように NWAM サービスの内部で使用されます。現在アクティブな NCP を変更するための適切なユーザーインタフェースは、`nwam-manager` GUI または `netadm` コマンドの `enable` サブコマンドです。

予約されている NCP 名 `DefaultFixed` を有効化すると、NWAM 自動構成管理が無効化され、従来の固定ネットワーク構成が有効化されます。

属性

属性についての詳細は、マニュアルページの `attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os

属性タイプ	属性値
インタフェースの安定性	流動的

関連項目

svcs(1)、dladm(1M)、ipadm(1M)、netcfg(1M)、netcfgd(1M)、nwamd(1M)、svadm(1M)、attribute

JDS/GNOME マニュアルページコレクションで利用可能な nwam-manager(1M) も参照してください。

注意事項

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

```
svc:/network/physical:default
```

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

名前

openssl – OpenSSL cryptographic and Secure Sockets Layer toolkit

機能説明

OpenSSL is a cryptography toolkit that implements the Secure Sockets Layer (SSLv2/v3) and Transport Layer Security (TLS v1) network protocols.

The following features are omitted from the binaries for issues including but not limited to patents, trademark, and US export restrictions: ECC, IDEA, MDC2, RC3, RC5, 4758_CCA Engine, AEP Engine, Atalla Engine, CHIL Engine, CSWIFT Engine, GMP Engine, NURON Engine, PadLock Engine, Sureware Engine, and UBSEC Engine.

The PKCS#11 Engine

A new PKCS#11 engine has been included with ENGINE name `pkcs11`. The engine was developed in Sun and is not integrated in the OpenSSL project.

The PKCS#11 engine is configured to use the Oracle Solaris Cryptographic Framework. See `cryptoadm(1M)` for configuration information.

The PKCS#11 engine can support the following set of mechanisms: `CKM_AES_CBC`, `CKM_AES_ECB`, `CKM_BLOWFISH_CBC`, `CKM_DES_CBC`, `CKM_DES_ECB`, `CKM_DES3_CBC`, `CKM_DES3_ECB`, `CKM_DSA`, `CKM_MD5`, `CKM_RC4`, `CKM_RSA_PKCS`, `CKM_RSA_X_509`, `CKM_SHA_1`, `CKM_SHA224`, `CKM_SHA256`, `CKM_SHA384`, `CKM_SHA512`, `CKM_SHA224_HMAC`, `CKM_SHA224_HMAC_GENERAL`, and `CKM_SHA224_KEY_DERIVATION`.

The set of mechanisms available depends on installed Crypto Framework providers. To see what mechanisms can be offloaded to the Cryptographic Framework through the PKCS#11 engine on a given machine, run the following command:

```
/usr/sfw/bin/openssl engine pkcs11 -vvv -t -c
```

Due to requirements of the PKCS#11 standard regarding `fork(2)` behavior, some applications that use the OpenSSL EVP interfaces and `fork()` with active `crypto` contexts might experience unexpected behavior.

The Dynamic Engine Support

The dynamic engine support has been enabled, which allows an external engine, in the form of a shared library, to be dynamically bound and used by an OpenSSL-based application.

Run the following command to see if the dynamic engine is supported:

```
$openssl engine dynamic
(dynamic) Dynamic engine loading support
```

The devcrypto Engine

This engine is implemented as a separate shared library, and it can be used by an OpenSSL application through the OpenSSL dynamic engine support. This engine was developed in Sun and is not integrated in the OpenSSL project.

Files for the `devcrypto` engine library are:

```
/lib/openssl/engines/libdevcrypto.so
/lib/openssl/engines/64/libdevcrypto.so
```

The devcrypto engine accesses only the kernel hardware providers from the Oracle Solaris Cryptographic Framework. To see the hardware provider information on a system, use the cryptoadm(1M) command.

The devcrypto engine supports the following set of mechanisms: CKM_AES_CBC, CKM_AES_CTR, CKM_AES_ECB, CKM_BLOWFISH_CBC, CKM_DES_CBC, CKM_DES_ECB, CKM_DES3_CBC, CKM_DES3_ECB, and CKM_RC4.

The set of mechanisms available depends on hardware providers installed and enabled in the Cryptographic Framework. To see what mechanisms are supported by the devcrypto engine on a particular machine, run the following command:

```
$openssl engine dynamic -pre SO_PATH:/lib/openssl/engines/libdevcrypto.so\
-pre LOAD -t -c
```

Using FIPS Mode

Currently a FIPS-140 certified mode is not available in Oracle Solaris.

Building an OpenSSL Application

To build an OpenSSL application, use the following cc command line options:

```
cc [ flag... ] file... -lcrypto -lssl [ library... ]
```

Accessing RSA Keys in PKCS#11 Keystores

OpenSSL can access RSA keys in PKCS#11 keystores using the following functions of the ENGINE API:

```
EVP_PKEY *ENGINE_load_private_key(ENGINE *e,
 const char *key_id, UI_METHOD *ui_method,
 void *callback_data)

EVP_PKEY *ENGINE_load_public_key(ENGINE *e,
 const char *key_id, UI_METHOD *ui_method,
 void *callback_data)
```

key_id, formerly for filenames only, can be now also set to a PKCS#11 URI. The EVP_PKEY structure is newly allocated and caller is responsible to free the structure later. To avoid clashes with existing filenames, file:// prefix for filenames is now also accepted but only when the PKCS#11 engine is in use. The PKCS#11 URI specification follows:

```
pkcs11:[token=<label>][:manuf=<label>][;serial=<label>]
   [;model=<label>][;object=<label>]
   [;objecttype=(public|private|cert)]
   [;passphrasedialog=(builtin|exec:<file>)]
```

The ordering of keywords is not significant. The PKCS#11 engine uses the keystore for the slot chosen for public key operations, which is metaslot on a standard configured machine. Currently, the PKCS#11 engine ignores the objecttype keyword. The only mandatory keyword is object which is the key object label. For information on how to use a different, possibly hardware, keystore with metaslot, see libpkcs11(3LIB).

The token PIN is provided by way of the passphrasedialog keyword and is either read from the terminal (builtin) or from the output of an external command (exec:<file>). The PIN

is used to log into the token and by default is deleted from the memory then. The keyword `pin` is intentionally not provided due to inherent security problems of possible use of a password in the process arguments.

Due to fork safety issues the application must re-login if the child continues to use the PKCS#11 engine. It is done inside of the engine automatically if fork is detected and in that case, `exec:<file>` option of the `passphrasedialog` keyword can be used. Alternatively, an environment variable `OPENSSL_PKCS11_PIN_CACHING_POLICY` can be used to allow the PIN to be cached in memory and reused in the child. It can be set to `none` which is the default, `memory` to store the PIN in memory, and `mlocked-memory` to keep the PIN in a locked page using `mlock(3C)`. `PRIV_PROC_LOCK_MEMORY` privilege is required in that case.

Sensitive parts of private keys are never read from the token to the process memory no matter whether the key is tagged with sensitive flag or not. The PKCS#11 engine uses the public components as a search key to get a PKCS#11 object handle to the private key.

To use the RSA keys by reference, high level API functions such as `RSA_public_decrypt()`, `EVP_PKEY_set1_RSA()`, or `EVP_SignInit()` must be used. Low level functions might go around the engine and fail to make use of the feature.

Additional Documentation

Extensive additional documentation for OpenSSL modules is available in the `/usr/share/man/man1openssl`, `/usr/share/man/man3openssl`, `/usr/share/man/man5openssl`, and `/usr/share/man/man7openssl` directories.

To view the license terms, attribution, and copyright for OpenSSL, run `pkg info --license library/security/openssl`.

使用例

例 1 Generating and Printing a Public Key

The following example generates and prints a public key stored in an already initilized PKCS#11 keystore. Notice the use of `-engine pkcs11` and `-inform e`.

```
$ pktool gencert keystore=pkcs11 label=mykey \
   subject="CN=test" keytype=rsa keylen=1024 serial=01
$ openssl rsa -in "pkcs11:object=mykey;passphrasedialog=builtin"\
   -pubout -text -engine pkcs11 -inform e
```

属性

See `attributes(5)` for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	library/security/openssl, library/security/openssl
Interface Stability	Volatile

関連項目 crle(1), cryptoadm(1M), libpkcs11(3LIB), attributes(5), privileges(5)

/usr/share/man/man1openssl/openssl.1openssl,
/usr/share/man/man1openssl/CRYPTO_num_locks.3openssl,
/usr/share/man/man3openssl/engine.3, /usr/share/man/man3openssl/evp.3

名前

pam_allow – PAM authentication, account, session and password management PAM module to allow operations

形式

`pam_allow.so.1`

機能説明

The `pam_allow` module implements all the PAM service module functions and returns `PAM_SUCCESS` for all calls. Opposite functionality is available in the `pam_deny(5)` module.

Proper Solaris authentication operation requires `pam_unix_cred(5)` be stacked above `pam_allow`.

The following options are interpreted:

debug Provides `syslog(3C)` debugging information at the `LOG_AUTH | LOG_DEBUG` level.

エラー

`PAM_SUCCESS` is always returned.

使用例

例 1 Allowing ssh none

The following example is a `pam.conf` fragment that illustrates how to allow the SSHv2 `userauth` of "none":

```
sshd-none  auth    required      pam_unix_cred.so.1
sshd-none  auth    sufficient    pam_allow.so.1
sshd-none  account sufficient    pam_allow.so.1
sshd-none  session sufficient    pam_allow.so.1
sshd-none  password sufficient   pam_allow.so.1
```

The equivalent configuration using `/etc/pam.d/` would be the following entries in `/etc/pam.d/sshd-none`:

```
auth    required      pam_unix_cred.so.1
auth    sufficient    pam_allow.so.1
account sufficient    pam_allow.so.1
session sufficient    pam_allow.so.1
password sufficient   pam_allow.so.1
```

例 2 Allowing Kiosk Automatic Login Service

The following example is a `pam.conf` fragment that illustrates how to allow `gdm` kiosk automatic login:

```
gdm-autologin auth  required    pam_unix_cred.so.1
gdm-autologin auth  sufficient  pam_allow.so.1
```

The equivalent configuration using `/etc/pam.d/` would be the following entries in `/etc/pam.d/gdm-autologin`:

```
auth  required    pam_unix_cred.so.1
auth  sufficient  pam_allow.so.1
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

libpam(3LIB), pam(3PAM), pam_sm(3PAM), syslog(3C), pam.conf(4), attributes(5), pam_deny(5), pam_unix_cred(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

This module is intended to be used to either allow access to specific services names, or to all service names not specified (by specifying it as the default service stack).

名前 pam_authtok_check – authentication and password management module

形式 `pam_authtok_check.so.1`

機能説明 `pam_authtok_check` provides functionality to the Password Management stack. The implementation of `pam_sm_chauthtok()` performs a number of checks on the construction of the newly entered password. `pam_sm_chauthtok()` is invoked twice by the PAM framework, once with flags set to `PAM_PRELIM_CHECK`, and once with flags set to `PAM_UPDATE_AUTHTOK`. This module only performs its checks during the first invocation. This module expects the current authentication token in the `PAM_OLDAUTHTOK` item, the new (to be checked) password in the `PAM_AUTHTOK` item, and the login name in the `PAM_USER` item. The checks performed by this module are:

length	The password length should not be less that the minimum specified in `/etc/default/passwd`.
circular shift	The password should not be a circular shift of the login name. This check may be disabled in `/etc/default/passwd`.
complexity	The password should contain at least the minimum number of characters described by the parameters `MINALPHA`, `MINNONALPHA`, `MINDIGIT`, and `MINSPECIAL`. Note that `MINNONALPHA` describes the same character classes as `MINDIGIT` and `MINSPECIAL` combined; therefore the user cannot specify both `MINNONALPHA` and `MINSPECIAL` (or `MINDIGIT`). The user must choose which of the two options to use. Furthermore, the `WHITESPACE` parameter determines whether whitespace characters are allowed. If unspecified `MINALPHA` is 2, `MINNONALPHA` is 1 and `WHITESPACE` is yes
variation	The old and new passwords must differ by at least the `MINDIFF` value specified in `/etc/default/passwd`. If unspecified, the default is 3. For accounts in name services which support password history checking, if prior history is defined, the new password must not match the prior passwords.
dictionary check	The password must not be based on a dictionary word. The list of words to be used for the site's dictionary can be specified with `DICTIONLIST`. It should contain a comma-separated list of filenames, one word per line. The database that is created from these files is stored in the directory named by `DICTIONDBDIR` (defaults to `/var/passwd`). See `mkpwdict(1M)` for information on pre-generating the database. If neither `DICTIONLIST` nor `DICTIONDBDIR` is specified, no dictionary check is made.
upper/lower case	The password must contain at least the minimum of upper- and lower-case letters specified by the `MINUPPER` and `MINLOWER` values in `/etc/default/passwd`. If unspecified, the defaults are 0.

maximum repeats — The password must not contain more consecutively repeating characters than specified by the MAXREPEATS value in /etc/default/passwd. If unspecified, no repeat character check is made.

The following option may be passed to the module:

force_check — If the PAM_NO_AUTHTOK_CHECK flag set, force_check ignores this flag. The PAM_NO_AUTHTOK_CHECK flag can be set to bypass password checks (see pam_chauthtok(3PAM)).

server_policy — If the account authority for the user, as specified by PAM_USER, is not files or NIS, and if server_policy is specified, this module does not perform any password-strength checks. Instead, it leaves it to the account authority to validate the new password against its own set of rules.

debug — syslog(3C) debugging information at the LOG_DEBUG level

戻り値

If the password in PAM_AUTHTOK passes all tests, PAM_SUCCESS is returned. If any of the tests fail, PAM_AUTHTOK_ERR is returned.

ファイル

/etc/default/passwd — See passwd(1) for a description of the contents.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

passwd(1), pam(3PAM), mkpwdict(1M), pam_chauthtok(3PAM), syslog(3C), libpam(3LIB), pam.conf(4), passwd(4), shadow(4), attributes(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前 pam_authtok_get – authentication and password management module

形式 `pam_authtok_get.so.1`

機能説明 The `pam_authtok_get` service module provides password prompting funtionality to the PAM stack. It implements `pam_sm_authenticate()` and `pam_sm_chauthtok()`, providing functionality to both the Authentication Stack and the Password Management Stack.

Authentication Service The implementation of `pam_sm_authenticate(3PAM)` prompts the user name if not set and then tries to get the authentication token from the pam handle. If the token is not set, it then prompts the user for a password and stores it in the `PAM` item `PAM_AUTHTOK`. This module is meant to be the first module on an authentication stack where users are to authenticate using a keyboard.

Password Management Service Due to the nature of the PAM Password Management stack traversal mechanism, the `pam_sm_chauthtok(3PAM)` function is called twice. Once with the `PAM_PRELIM_CHECK` flag, and one with the `PAM_UPDATE_AUTHTOK` flag.

In the first (`PRELIM`) invocation, the implementation of `pam_sm_chauthtok(3PAM)` moves the contents of the `PAM_AUTHTOK` (current authentication token) to `PAM_OLDAUTHTOK`, and subsequentially prompts the user for a new password. This new password is stored in `PAM_AUTHTOK`.

If a previous module has set `PAM_OLDAUTHTOK` prior to the invocation of pam_authtok_get, this module turns into a `NO-OP` and immediately returns `PAM_SUCCESS`.

In the second (`UPDATE`) invocation, the user is prompted to Re-enter his password. The pam_sm_chauthtok implementation verifies this reentered password with the password stored in `PAM_AUTHTOK`. If the passwords match, the module returns `PAM_SUCCESS`.

The following option can be passed to the module:

debug `syslog(3C)` debugging information at the `LOG_DEBUG` level

エラー The authentication service returns the following error codes:

`PAM_SUCCESS` Successfully obtains authentication token

`PAM_SYSTEM_ERR` Fails to retrieve username, username is `NULL` or empty

The password management service returns the following error codes:

`PAM_SUCCESS` Successfully obtains authentication token

`PAM_AUTHTOK_ERR` Authentication token manipulation error

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

pam(3PAM), pam_authenticate(3PAM), syslog(3C), libpam(3LIB), pam.conf(4), attributes(5), pam_authtok_check(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_authtok_store – password management module

形式

`pam_authtok_store.so.1`

機能説明

`pam_authtok_store` provides functionality to the PAM password management stack. It provides one function: `pam_sm_chauthtok()`.

When invoked with flags set to `PAM_UPDATE_AUTHTOK`, this module updates the authentication token for the user specified by `PAM_USER`.

The authentication token `PAM_OLDAUTHTOK` can be used to authenticate the user against repositories that need updating (NIS, LDAP). After successful updates, the new authentication token stored in `PAM_AUTHTOK` is the user's valid password.

This module honors the `PAM_REPOSITORY` item, which, if set, specifies which repository is to be updated. If `PAM_REPOSITORY` is unset, it follows the `nsswitch.conf(4)`.

The following option can be passed to the module:

`debug` `syslog(3C)` debugging information at the `LOG_DEBUG` level

`server_policy` If the account authority for the user, as specified by `PAM_USER`, is a server, do not encrypt the authentication token before updating.

エラー

`PAM_SUCCESS` Successfully obtains authentication token

`PAM_SYSTEM_ERR` Fails to get username, service name, old password or new password, user name null or empty, or password null.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `pam_chauthtok(3PAM)`, `syslog(3C)`, `libpam(3LIB)`, `pam.conf(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`, `pam_unix_auth(5)`, `pam_unix_session(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

If the `PAM_REPOSITORY` *item_type* is set and a service module does not recognize the type, the service module does not process any information, and returns `PAM_IGNORE`. If the `PAM_REPOSITORY` *item_type* is not set, a service module performs its default action.

名前

pam_deny – PAM authentication, account, session and password management PAM module to deny operations

形式

`pam_deny.so.1`

機能説明

The `pam_deny` module implements all the PAM service module functions and returns the module type default failure return code for all calls.

The following options are interpreted:

debug `syslog(3C)` debugging information at the `LOG_AUTH|LOG_DEBUG` levels

エラー

The following error codes are returned:

`PAM_ACCT_EXPIRED` If `pam_sm_acct_mgmt` is called.

`PAM_AUTH_ERR` If `pam_sm_authenticate` is called.

`PAM_AUTHOK_ERR` If `pam_sm_chauthtok` is called.

`PAM_CRED_ERR` If `pam_sm_setcred` is called.

`PAM_SESSION_ERR` If `pam_sm_open_session` or `pam_sm_close_session` is called.

使用例

例 1 Disallowing `ssh` none authentication

The following example is a `pam.conf` fragment that illustrates how to deny the SSHv2 `userauth` of "none":

```
sshd-none       auth       requisite    pam_deny.so.1
sshd-none       account    requisite    pam_deny.so.1
sshd-none       session    requisite    pam_deny.so.1
sshd-none       password   requisite    pam_deny.so.1
```

The equivalent configuration in `/etc/pam.d/` would be the following entries in `/etc/pam.d/sshd-none:`

```
auth       requisite    pam_deny.so.1
account    requisite    pam_deny.so.1
session    requisite    pam_deny.so.1
password   requisite    pam_deny.so.1
```

例 2 Disallowing any service not explicitly defined

The following example is a `pam.conf` fragment that illustrates how to deny any PAM service which is not explicitly defined in the PAM configuration:

```
other           auth       requisite    pam_deny.so.1
other           account    requisite    pam_deny.so.1
other           session    requisite    pam_deny.so.1
other           password   requisite    pam_deny.so.1
```

例 2 Disallowing any service not explicitly defined *(続き)*

The equivalent configuration in `/etc/pam.d/` would be the following entries in `/etc/pam.d/other`:

```
auth       requisite    pam_deny.so.1
account    requisite    pam_deny.so.1
session    requisite    pam_deny.so.1
password   requisite    pam_deny.so.1
```

属性

See `attributes(5)` for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	MT-Safe with exceptions

関連項目

`su(1M)`, `libpam(3LIB)`, `pam(3PAM)`, `pam_sm_authenticate(3PAM)`, `syslog(3C)`, `pam.conf(4)`, `nsswitch.conf(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`, `pam_unix_auth(5)`, `pam_unix_session(5)`, `privileges(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

The `pam_deny` module is intended to deny access to a specified service. The `other` service name may be used to deny access to services not explicitly specified.

名前

pam_dhkeys – authentication Diffie-Hellman keys management module

形式

`pam_dhkeys.so.1`

機能説明

The `pam_dhkeys.so.1` service module provides functionality to two PAM services: Secure RPC authentication and Secure RPC authentication token management.

Secure RPC authentication differs from regular Unix authentication because ONC RPCs use Secure RPC as the underlying security mechanism.

The following options may be passed to the module:

debug `syslog(3C)` debugging information at `LOG_DEBUG` level

nowarn Turn off warning messages

Authentication Services

If the user has Diffie-Hellman keys, `pam_sm_authenticate()` establishes secret keys for the user specified by the `PAM_USER` (equivalent to running `keylogin(1)`), using the authentication token found in the `PAM_AUTHTOK` item. If `pam_sm_setcred()` is called with `PAM_ESTABLISH_CRED` and the user's secure `RPC` credentials need to be established, these credentials are set. This is equivalent to running `keylogin`(1).

If the credentials could not be set and `PAM_SILENT` is not specified, a diagnostic message is displayed. If `pam_setcred()` is called with `PAM_DELETE_CRED`, the user's secure RPC credentials are unset. This is equivalent to running `keylogout(1)`.

`PAM_REINITIALIZE_CRED` and `PAM_REFRESH_CRED` are not supported and return `PAM_IGNORE`.

Authentication Token Management

The `pam_sm_chauthtok()` implementation checks whether the old login password decrypts the users secret keys. If it doesn't this module prompts the user for an old Secure RPC password and stores it in a pam data item called `SUNW_OLDRPCPASS`. This data item can be used by the store module to effectively update the users secret keys.

エラー

The authentication service returns the following error codes:

`PAM_SUCCESS`	Credentials set successfully.
`PAM_IGNORE`	Credentials not needed to access the password repository.
`PAM_USER_UNKNOWN`	`PAM_USER` is not set, or the user is unknown.
`PAM_AUTH_ERR`	No secret keys were set. `PAM_AUTHTOK` is not set, no credentials are present or there is a wrong password.
`PAM_BUF_ERR`	Module ran out of memory.

The authentication token management returns the following error codes:

`PAM_SUCCESS`	Old `rpc` password is set in `SUNW_OLDRPCPASS`
`PAM_USER_UNKNOWN`	User in `PAM_USER` is unknown.

PAM_AUTHTOK_ERR	User did not provide a password that decrypts the secret keys.
PAM_BUF_ERR	Module ran out of memory.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

keylogin(1), keylogout(1), pam(3PAM), pam_authenticate(3PAM), pam_chauthtok(3PAM), pam_setcred(3PAM), pam_get_item(3PAM), pam_set_data(3PAM), pam_get_data(3PAM), syslog(3C), libpam(3LIB), pam.conf(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_dial_auth – authentication management PAM module for dialups

形式

`pam_dial_auth.so.1`

機能説明

The `pam_dial_auth` module implements `pam_sm_authenticate(3PAM)` which authenticates the user according to the `dialups(4)` and `d_passwd(4)` files configuration.

Authentication service modules must implement both `pam_sm_authenticate()` and `pam_sm_setcred()`. `pam_sm_setcred()` in this module always returns `PAM_IGNORE`.

The value of the `PAM_TTY` item is checked against entries in `dialups(4)`. If there is a match, the user's shell is compared against entries in `d_passwd(4)`. If there is a matching entry, the user is prompted for a password which is validated against the entry found.

The following option may be passed in to this service module:

`debug` `syslog(3C)` debugging information at `LOG_DEBUG` level.

エラー

If `dialups(4)` is not present, `PAM_IGNORE` is returned. Upon successful completion of `pam_sm_authenticate()`, `PAM_SUCCESS` is returned. The following error codes are returned upon error:

`PAM_AUTH_ERR`	Authentication failure.
`PAM_SERVICE_ERR`	Error in the calling service, `PAM_TTY` is not set.
`PAM_SYSTEM_ERR`	System error (`d_passwd(4)` is not present).
`PAM_USER_UNKNOWN`	No account is present for *user*.

属性

See `attributes(5)` for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `pam_sm_authenticate(3PAM)`, `d_passwd(4)`, `dialups(4)`, `libpam(3LIB)`, `pam.conf(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`, `pam_unix_auth(5)`, `pam_unix_session(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own `PAM` handle.

名前

pam_krb5 – authentication, account, session, and password management PAM modules for Kerberos V5

形式

`/usr/lib/security/pam_krb5.so.1`

機能説明

The Kerberos V5 service module for PAM provides functionality for all four PAM modules: authentication, account management, session management, and password management. The service module is a shared object that can be dynamically loaded to provide the necessary functionality upon demand. Its path is specified in the PAM configuration file.

Kerberos Authentication Module

The Kerberos V5 authentication component provides functions to verify the identity of a user, `pam_sm_authenticate()`, and to manage the Kerberos credentials cache, `pam_sm_setcred()`.

`pam_sm_authenticate()` authenticates a user principal through the Kerberos authentication service. If the authentication request is successful, the authentication service sends a ticket-granting ticket (TGT) back to the service module, which then verifies that the TGT came from a valid Key Distribution Center (KDC) by attempting to get a service ticket for the local host service. For this to succeed, the local host's `keytab` file (`/etc/krb5/krb5.keytab`) must contain the entry for the local host service. For example, in the file `host/`*hostname.com*`@`*REALM*, *hostname.com* is the fully qualified local hostname and *REALM* is the default realm of the local host as defined in `/etc/krb5/krb5.conf`. If the host entry is not found in the keytab file, the authentication fails. Administrators can optionally disable this "strict" verification by setting "`verify_ap_req_nofail = false`" in `/etc/krb5/krb5.conf`. See `krb5.conf(4)` for more details on this option. This allows TGT verification to succeed in the absence of a `keytab` host principal entry.

If `pam_sm_authenticate()` is called and the `pkinit` module option is set, the Kerberos V5 authentication module tries to do `PKINIT` authentication, assuming that both the system and the `KDC` are configured to support this type of authentication. This form of authentication uses a user's certificate and private key to acquire the user's initial Kerberos credential (`TGT`). One of the keystore formats supported is `PKCS11` which supports use of any `PKCS11` compatible keystore capable of storing the required credential and private key needed for `PKINIT` authentication (`PKCS11` compatible smartcards are an example). See `krb5.conf(4)` for more details on `PKINIT` configuration. This form of authentication is typically useful for services where the system on which the `auth` stack is being processed has access to the user's certificate and private key.

If `pam_sm_authenticate()` is called and the `pkinit` module option is not set then the Kerberos V5 authentication module does password based authentication.

In either case, if the `PAM_AUTHTOK` password item has been set when `pam_sm_authenticate()` is called, which is the case when `pam_krb5` is stacked after `pam_authtok_get` in the `auth` stack, the Kerberos V5 authentication module uses that `PAM_AUTHTOK` password for either `PKINIT` or password based Kerberos authentication.

If the `PAM_USER` item is not set `pam_krb5` with the `pkinit` option prompts for and set that item.

If the PAM_AUTHTOK password item has not been set when pam_sm_authenticate() is called, which is the case when pam_krb5 is stacked before pam_authtok_get in the auth stack, and the pkinit option is present the Kerberos V5 authentication module allows the Kerberos pkinit preauth plugin to prompt for whatever information is needed to perform PKINIT (typically this is for the user's PIN). No PAM items are set by way of this prompting. See krb5.conf(4) for more information on PKINIT configuration options.

The pam_krb5 module sets the KRB5CCNAME shell environment variable upon successful authentication or password change to FILE:/tmp/krb5cc_*uid* where *uid* is the UID of the user that pam_krb5 authenticated. KRB5CCNAME is documented in krb5envvar(5).

If it is desirable to initially have the Kerberos V5 authentication module try PKINIT Kerberos authentication and fall back to password based Kerberos authentication then either the sufficient or optional control flags must be provided for the instance of pam_krb5 with the pkinit module option set and another instance of pam_krb5 without the pkinit module option must be stacked below pam_authtok_get. If there are PAM modules other than pam_krb5 that must be evaluated below pam_authtok_get then the control flag should be set to optional for the instance of pam_krb5 with the pkinit module option set otherwise the control flag should be set to sufficient.

Only two instances of pam_krb5 are supported in a auth stack.

pam_sm_authenticate(3PAM) can be passed the following flag:

PAM_DISALLOW_NULL_AUTHTOK
: This flag is ignored. The Kerberos authentication mechanism does not allow an empty password string by default.

pam_sm_setcred() creates and modifies the user's credential cache. This function initializes the user's credential cache, if it does not already exist, and stores the initial credentials for later use by Kerberos network applications. The following flags can be set in the flags field. They are best described by their effect on the user's credential cache.

PAM_ESTABLISH_CRED
: Stores the initial credentials in the user's credential cache so that the user can access Kerberos network services. If a successful authentication pass was made, the new credentials are stored in the credential cache, overwriting any existing credentials that were previously stored. If an unsuccessful authentication pass was made, PAM_CRED_UNAVAIL is returned.

PAM_DELETE_CRED
: This flag has no effect on the credential cache and always returns PAM_SUCCESS. The credential cache is not deleted because there is no accurate method to determine if the credentials are needed by another process. The credential cache can be deleted with the kdestroy(1) command.

PAM_REINITIALIZE_CRED
: Deletes the user's existing credential cache, if it exists, and creates a new credential cache. The new credentials are stored in the new cache and the user's ticket lifetime and renewable life time values are reset.

PAM_REFRESH_CRED
: Does not require a previous authentication pass, but if a successful one is made, the new credentials are stored in the credential cache. If a previous authentication pass was not made or was unsuccessful, an attempt to renew the existing credentials is made. This function fails if the user's renewable ticket lifetime is expired.

The following options can be passed to the Kerberos V5 authentication module:

debug
: Provides syslog(3C) debugging information at LOG_DEBUG level.

nowarn
: Turns off warning messages.

pkinit
: Indicates that the Kerberos V5 authentication module should try Kerberos PKINIT authentication instead of the default password based Kerberos authentication.

Kerberos V5 Account Management Module

The Kerberos account management component provides a function to perform account management, pam_sm_acct_mgmt(). This function checks to see if the pam_krb5 authentication module has noted that the user's password has not expired. The following options can be passed in to the Kerberos V5 account management module:

debug
: Provides syslog(3C) debugging information at LOG_DEBUG level

nowarn
: Turns off warning messages. Also, does not query KDC for impending password expiration information used to warn the user.

Kerberos V5 Session Management Module

The Kerberos V5 session management component provides functions to initiate pam_sm_open_session() and terminate pam_sm_close_session() Kerberos sessions. For Kerberos V5, both pam_sm_open_session and pam_sm_close_session() are null functions, returning PAM_IGNORE.

Kerberos V5 Password Management Module

The Kerberos V5 password management component provides a function to change passwords, pam_sm_chauthtok(), in the Key Distribution Center (KDC) database.

If the Kerberos V5 authentication module used PKINIT authentication in the auth stack then the Kerberos V5 password management module returns PAM_IGNORE in the following cases:

- The new password is NULL.
- The old password is NULL.
- Verification of the old password fails.

The rationale behind this is that the KDC can not allow a PKINIT user to change/set a password since the user can be expected to use PKINIT only. If all of the cases above are false the Kerberos V5 password management module tries to change the user's password in the KDC database.

If the KDC only supports PKINIT authentication then the Kerberos V5 password management module should not be present in any password stacks.

Related to PKINIT the Kerberos V5 password management module does not support changing the key store PIN used to access a user's private key and certificate.

The following flags can be passed to pam_sm_chauthtok(3PAM):

PAM_CHANGE_EXPIRED_AUTHTOK
: The password service should only update the user's Kerberos password if it is expired. Otherwise, this function returns PAM_IGNORE. The default behaviour is to always change the user's Kerberos password.

PAM_PRELIM_CHECK
: This is a null function that always returns PAM_IGNORE.

PAM_UPDATE_AUTHTOK
: This flag is necessary to change the user's Kerberos password. If this flag is not set, pam_krb5 returns PAM_SYSTEM_ERR.

The following option can be passed to the Kerberos V5 password module:

debug Provides syslog(3C) debugging information at LOG_DEBUG level.

エラー

The following error codes are returned for pam_sm_authenticate():

PAM_AUTH_ERR	Authentication failure
PAM_BUF_ERR	Memory buffer error.
PAM_IGNORE	The user is "root" and the root key exists in the default keytab.
PAM_SUCCESS	Successfully obtained Kerberos credentials .
PAM_SYSTEM_ERR	System error.
PAM_USER_UNKNOWN	An unknown Kerberos principal was requested.

The following error codes are returned for pam_sm_setcred():

PAM_AUTH_ERR	Authentication failure.
PAM_BUF_ERR	Memory buffer error.
PAM_IGNORE	The user is "root" and the root key exists in the default keytab.
PAM_SYSTEM_ERR	System error.
PAM_SUCCESS	Successfully modified the Kerberos credential cache.

The following error codes are returned for pam_sm_acct_mgmt():

PAM_AUTH_ERR	Authentication failure.

PAM_IGNORE	Kerberos service module pam_sm_authenticate() was never called, or the user is "root" and the root key exists in the default keytab.
PAM_NEW_AUTHTOK_REQD	Obtain new authentication token from the user.
PAM_SERVICE_ERR	Error in underlying service module.
PAM_SUCCESS	Kerberos principal account is valid.
PAM_SYSTEM_ERR	System error.
PAM_USER_UNKNOWN	An unknown Kerberos principal was requested.

The following error code is returned for pam_sm_open_session() and pam_sm_close_session():

PAM_IGNORE These two functions are null functions in pam_krb5:

The following error codes are returned for pam_sm_chauthtok():

PAM_AUTH_ERR	Authentication failure.
PAM_IGNORE	The user has not been authenticated by Kerberos service module pam_sm_authenticate(), or the user is "root" and the root key exists in the default keytab.
PAM_NEW_AUTHTOK_REQD	User's Kerberos password has expired.
PAM_SERVICE_ERR	Error in module. At least one input parameter is missing.
PAM_SYSTEM_ERR	System error.
PAM_USER_UNKNOWN	An unknown Kerberos principal was requested.
PAM_SUCCESS	Successfully changed the user's Kerberos password.

使用例

例 1 Authenticating Users Through Kerberos as First Choice Using Password-based Authentication

The following is an excerpt of a sample pam.conf configuration file that authenticates users through the Kerberos authentication service and authenticates through the Unix login only if the Kerberos authentication fails. This arrangement is helpful when a majority of the users are networked by means of Kerberos and when there are only a few non-Kerberos type user accounts, such as root. The service illustrated below is for gdm.

```
gdm auth requisite          pam_authtok_get.so.1
gdm auth required           pam_dhkeys.so.1
gdm auth required           pam_unix_cred.so.1
gdm auth sufficient         pam_krb5.so.1
gdm auth required           pam_unix_auth.so.1
```

例 1 Authenticating Users Through Kerberos as First Choice Using Password-based Authentication (続き)

These changes should not be made to the existing `krlogin`, `krsh`, and `ktelnet` service entries. Those services require Kerberos authentication, so using a seemingly sufficient control flag would not provide the necessary functionality for privacy and integrity. There should be no need to change those entries.

The following entries check for password expiration when dealing with Kerberos and Unix password aging policies:

```
other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account required        pam_krb5.so.1
```

The following entries would change the Kerberos password of the user and continue to change the Unix login password only if the Kerberos password change had failed:

```
other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password sufficient     pam_krb5.so.1
other   password required       pam_authtok_store.so.1
```

例 2 Authenticating Users Through Kerberos Only Using Password-based Authentication

The following example allows authentication only to users that have Kerberos-based accounts.

```
gdm auth requisite          pam_authtok_get.so.1
gdm auth required           pam_dhkeys.so.1
gdm auth required           pam_unix_cred.so.1
gdm auth required           pam_krb5.so.1
```

Typically, you would have another service specified in the `pam.conf` file that would allow local users, such as database, web server, system administrator accounts, to log in to the host machine. For example, the service name “login” could be used for these users. These users should not belong to any roles.

The rest of the module types look similar to that shown in the previous example:

```
other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account required        pam_krb5.so.1
```

With binding specified in the following, it is important that non-Kerberos users specify the repository in which they reside using the `-r` option with the `passwd(1)` command. This configuration is also based on the assumptions that:

- Kerberos users maintain only their Kerberos passwords;

例 2 Authenticating Users Through Kerberos Only Using Password-based Authentication (続き)

- changing their Unix password is not necessary, given that they are authenticated only through their Kerberos passwords when logging in.

```
other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password binding        pam_krb5.so.1
```

例 3 Authenticating Through Kerberos Optionally Using Password-based Authentication

This configuration is helpful when the majority of users are non-Kerberos users and would like to authenticate through Kerberos if they happened to exist in the Kerberos database. The effect of this is similar to users voluntarily executing `kinit(1)` after they have successfully logged in:

```
gdm auth requisite          pam_authtok_get.so.1
gdm auth required           pam_dhkeys.so.1
gdm auth required           pam_unix_cred.so.1
gdm auth required           pam_unix_auth.so.1
gdm auth optional           pam_krb5.so.1
```

The rest of the configuration is as follows:

```
other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account optional        pam_krb5.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password required       pam_authtok_store.so.1
other   password optional       pam_krb5.so.1
```

Non-Kerberos users should specify their respective repositories by using the `-r` option when changing their password with the `passwd(1)` command.

例 4 Authenticating Users Through Kerberos PKINIT as First Choice

The following is an excerpt of a sample `pam.conf` configuration file that authenticates users through the Kerberos authentication service and authenticates through the Unix login only if the Kerberos authentication (using `PKINIT`) fails. This arrangement is helpful when a majority of the users are networked by means of Kerberos and when there are only a few non-Kerberos type user accounts, such as root. The service illustrated below is for login. The user is prompted once for the PIN by `pam_krb5`.

```
login auth required           pam_unix_cred.so.1
```

例 4 Authenticating Users Through Kerberos PKINIT as First Choice *(続き)*

```
login auth sufficient         pam_krb5.so.1 pkinit
login auth requisite          pam_authtok_get.so.1
login auth required           pam_dhkeys.so.1
login auth required           pam_unix_auth.so.1
```

例 5 Authenticating Users Through Kerberos PKINIT Only

The following example allows authentication only to users that have kerberos-based accounts requiring `PKINIT` authentication.

```
login auth required           pam_unix_cred.so.1
login auth required           pam_krb5.so.1 pkinit
```

例 6 Authenticating Users Through Kerberos PKINIT Optionally

The following example allows users to acquire a Kerberos credential using `PKINIT` authentication if they have a Kerberos account. Whether `pam_krb5` succeeds or fails the user must provide their Unix password to login.

```
login auth required           pam_unix_cred.so.1
login auth optional           pam_krb5.so.1 pkinit
login auth requisite          pam_authtok_get.so.1
login auth required           pam_unix_auth.so.1
```

例 7 Authenticating Users Through Kerberos PKINIT as a Requirement

The following example allows users to login if `pam_krb5` is able to acquire a Kerberos credential using `PKINT` authentication and in addition must provide their Unix password to `pam_unix_auth`.

```
login auth required           pam_unix_cred.so.1
login auth required           pam_krb5.so.1 pkinit
login auth requisite          pam_authtok_get.so.1
login auth required           pam_unix_auth.so.1
```

例 8 Authenticating Users Through Kerberos PKINIT as a Requirement

The following example allows users to login using their `PAM_AUTHTOK` password acquired by `pam_authtok_get`. This password is used by `pam_krb5` to try `PKINIT` authentication and is also used by `pam_unix_auth` to authenticate the user using the user's Unix account. If `PKINIT` requires a password/PIN that differs from the user's Unix password then `pam_krb5` must be stacked above `pam_authtok_get`.

```
login auth required           pam_unix_cred.so.1
login auth requisite          pam_authtok_get.so.1
login auth required           pam_krb5.so.1 pkinit
login auth required           pam_unix_auth.so.1
```

例 9 Authenticating Users Through Kerberos PKINIT with a Fall Back to Password-based krb auth

The following example allows users to acquire a Kerberos credential using `PKINIT` authentication or using password based authentication if `PKINIT` fails. If `PKINIT` succeeds the user is not prompted for their password. If `pam_krb5` `PKINIT` succeeds, the second instance of `pam_krb5` does not try password authentication and returns success. If `PKINIT` fails the user is prompted for their Kerberos password.

```
login auth required           pam_unix_cred.so.1
login auth sufficient         pam_krb5.so.1 pkinit
login auth requisite          pam_authtok_get.so.1
login auth required           pam_krb5.so.1
```

例 10 Authenticating Users Through Kerberos Requiring Users to Authenticate Either through Kerberos PKINIT or Fall Back to Password-based krb auth

The following example allows users to acquire a Kerberos credential using `PKINIT` authentication or using password based authentication if `PKINIT` fails. If `pam_krb5` `PKINIT` succeeds, the second instance of `pam_krb5` does not try password authentication and returns ignore. If `pam_krb5` `PKINIT` fails the second instance of `pam_krb5` tries password based authentication and return success or failure.

```
login auth required           pam_unix_cred.so.1
login auth optional           pam_krb5.so.1 pkinit
login auth requisite          pam_authtok_get.so.1
login auth required           pam_krb5.so.1
login auth required           pam_dhkeys.so.1
login auth required           pam_unix_auth.so.1
```

例 11 Authenticating Users Through Kerberos Requiring Users to Authenticate Either through Kerberos PKINIT or Fall Back to pam_pkcs11

The following example allows users to acquire a Kerberos credential using `PKINIT` authentication or if that fails use `pam_pkcs11` to validate the user's PIN using their certificate and private key.

```
login auth required           pam_unix_cred.so.1
login auth sufficient         pam_krb5.so.1 pkinit
login auth sufficient         pam_pkcs11.so
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

`kdestroy(1)`, `kinit(1)`, `passwd(1)`, `ktkt_warnd(1M)`, `libpam(3LIB)`, `pam(3PAM)`, `pam_sm(3PAM)`, `pam_sm_acct_mgmt(3PAM)`, `pam_sm_authenticate(3PAM)`, `pam_sm_chauthtok(3PAM)`, `pam_sm_close_session(3PAM)`, `pam_sm_open_session(3PAM)`,

pam_sm_setcred(3PAM), syslog(3C), krb5.conf(4), pam.conf(4), attributes(5), kerberos(5), krb5envvar(5), pam_krb5_migrate(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

On successful acquisition of initial credentials (ticket-granting ticket), ktkt_warnd(1M) is notified, to alert the user when the initial credentials are about to expire.

名前

pam_krb5_migrate – authentication PAM module for the KerberosV5 auto-migration of users feature

形式

`/usr/lib/security/pam_krb5_migrate.so.1`

機能説明

The KerberosV5 auto-migrate service module for PAM provides functionality for the PAM authentication component. The service module helps in the automatic migration of `PAM_USER` to the client's local Kerberos realm, using `PAM_AUTHTOK` (the PAM authentication token associated with `PAM_USER`) as the new Kerberos principal's password.

KerberosV5 Auto-migrate Authentication Module

The KerberosV5 auto-migrate authentication component provides the `pam_sm_authenticate(3PAM)` function to migrate a user who does not have a corresponding `krb5` principal account to the default Kerberos realm of the client.

`pam_sm_authenticate(3PAM)` uses a host-based client service principal, present in the local `keytab` (`/etc/krb5/krb5.keytab`) to authenticate to `kadmind(1M)` (defaults to the `host/nodename.fqdn` service principal), for the principal creation operation. Also, for successful creation of the `krb5` user principal account, the host-based client service principal being used needs to be assigned the appropriate privilege on the master KDC's `kadm5.acl(4)` file. `kadmind(1M)` checks for the appropriate privilege and validates the user password using PAM by calling `pam_authenticate(3PAM)` and `pam_acct_mgmt(3PAM)` for the `k5migrate` service.

If migration of the user to the KerberosV5 infrastructure is successful, the module will inform users about it by means of a `PAM_TEXT_INFO` message, unless instructed otherwise by the presence of the `quiet` option.

The authentication component always returns `PAM_IGNORE` and is meant to be stacked in the PAM configuration (see `pam.conf(4)`) with a requirement that it be listed below `pam_authtok_get(5)` in the authentication stack. Also, if `pam_krb5_migrate` is used in the authentication stack of a particular service, it is mandatory that `pam_krb5(5)` be listed in the PAM account stack of that service for proper operation (see EXAMPLES).

オプション

The following options can be passed to the KerberosV5 auto-migrate authentication module:

`debug`	Provides `syslog(3C)` debugging information at `LOG_DEBUG` level.
`client_service=`<*service name*>	Name of the service used to authenticate to `kadmind(1M)` defaults to `host`. This means that the module uses `host/`<*nodename.fqdn*> as its client service principal name, KerberosV5 user principal creation operation or <*service*>/<*nodename.fqdn*> if this option is provided.
`quiet`	Do not explain KerberosV5 migration to the user.

This has the same effect as passing the PAM_SILENT flag to pam_sm_authenticate(3PAM) and is useful where applications cannot handle PAM_TEXT_INFO messages.

If not set, the authentication component will issue a PAM_TEXT_INFO message after creation of the Kerberos V5 principal, indicating that it has done so.

expire_pw

Causes the creation of KerberosV5 user principals with password expiration set to now (current time).

使用例

例 1 Sample PAM entries for the pam_krb5_migrate.so.1 module

The following pam.conf(4) entries demonstrate the use of the pam_krb5_migrate.so.1 module:

```
login       auth requisite          pam_authtok_get.so.1
login       auth required           pam_dhkeys.so.1
login       auth required           pam_unix_cred.so.1
login       auth sufficient         pam_krb5.so.1
login       auth requisite          pam_unix_auth.so.1
login       auth optional           pam_krb5_migrate.so.1 expire_pw
login       auth required           pam_dial_auth.so.1

other   account requisite       pam_roles.so.1
other   account required        pam_krb5.so.1
other   account required        pam_unix_account.so.1
```

The equivalent PAM configuration in /etc/pam.d/ would be the following entries in /etc/pam.d/login:

```
auth    requisite          pam_authtok_get.so.1
auth    required           pam_dhkeys.so.1
auth    required           pam_unix_cred.so.1
auth    sufficient         pam_krb5.so.1
auth    requisite          pam_unix_auth.so.1
auth    optional           pam_krb5_migrate.so.1 expire_pw
auth    required           pam_dial_auth.so.1
```

and the following entries in /etc/pam.d/other:

```
account requisite             pam_roles.so.1
account required              pam_krb5.so.1
account required              pam_unix_account.so.1
```

The pam_krb5_migrate module can generally be present on the authentication stack of any service where the application calls pam_sm_authenticate(3PAM) and an authentication token (in the preceding example, the authentication token would be the user's Unix password) is available for use as a Kerberos V5 password.

例 2 Sample Entries from kadm5.acl

The following entries from kadm5.acl(4) permit or deny privileges to the host client service principal:

```
host/*@ACME.COM U root
host/*@ACME.COM ui *
```

The preceding entries permit the pam_krb5_migrate add privilege to the host client service principal of any machine in the ACME.COM KerberosV5 realm, but denies the add privilege to all host service principals for addition of the root user account.

例 3 Sample PAM entries for the Master KDC

The /etc/pam.conf entries below enable kadmind(1M) on the master KDC to use the k5migrate PAM service in order to validate Unix user passwords for accounts that require migration to the Kerberos realm.

```
k5migrate       auth    required        pam_unix_auth.so.1
k5migrate       account required        pam_unix_account.so.1
```

The equivalent PAM configuration in /etc/pam.d/ would be the following entries in /etc/pam.d/k5migrate:

```
auth    required        pam_unix_auth.so.1
account required        pam_unix_account.so.1
```

属性

See attributes(5) for a description of the following attribute:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

関連項目

kadmind(1M), syslog(3C), pam_authenticate(3PAM), pam_acct_mgmt(3PAM), pam_sm_authenticate(3PAM), kadm5.acl(4), pam.conf(4), attributes(5), pam_authtok_get(5), pam_krb5(5)

名前

pam_ldap – authentication and account management PAM module for LDAP

形式

/usr/lib/security/pam_ldap.so.1

機能説明

The pam_ldap module implements pam_sm_authenticate(3PAM) and pam_sm_acct_mgmt(3PAM), the functions that provide functionality for the PAM authentication and account management stacks. The pam_ldap module ties the authentication and account management functionality to the functionality of the supporting LDAP server. For authentication, pam_ldap can authenticate the user directly to any LDAP directory server by using any supported authentication mechanism, such as DIGEST-MD5. However, the account management component of pam_ldap will work only with the Sun Java System Directory Server. The server's user account management must be properly configured before it can be used by pam_ldap. Refer to the Sun Java System Directory Server Administration Guide for information on how to configure user account management, including password and account lockout policy.

pam_ldap must be used in conjunction with the modules that support the UNIX authentication, password, and account management, which are pam_authtok_get(5), pam_passwd_auth(5), pam_unix_account(5), and pam_unix_auth(5). pam_ldap is designed to be stacked directly below these modules. If other modules are designed to be stacked in this manner, the modules can be stacked below the pam_ldap module. The 使用例 section shows how the UNIX modules are stacked with pam_ldap. When stacked together, the UNIX modules are used to control local accounts, such as root. pam_ldap is used to control network accounts, that is, LDAP users. For the stacks to work, pam_unix_auth, pam_unix_account, and pam_passwd_auth must be configured with the binding control flag and the server_policy option. This configuration allows local account override of a network account.

LDAP Authentication Module

The LDAP authentication module verifies the identity of a user. The pam_sm_authenticate(3PAM) function uses the password entered by the user to attempt to authenticate to the LDAP server. If successful, the user is authenticated. See NOTES for information on password prompting.

The authentication method used is either defined in the client profile , or the authentication method is configured by using the ldapclient(1M) command. To determine the authentication method to use, this module first attempts to use the authentication method that is defined, for service pam_ldap, for example, serviceAuthenticationMethod:pam_ldap:sasl/DIGEST-MD5. If no authentication method is defined, pam_ldap uses the default authentication method. If neither are set, the authentication fails. This module skips the configured authentication method if the authentication method is set to none.

The following options can be passed to the LDAP service module:

debug　syslog(3C) debugging information at LOG_DEBUG level.

nowarn　Turn off warning messages.

These options are case sensitive and must be used exactly as presented here.

LDAP Account Management Module

The LDAP account management module validates the user's account. The pam_sm_acct_mgmt(3PAM) function authenticates to the LDAP server to verify that the user's password has not expired, or that the user's account has not been locked. In the event that there is no user authentication token (PAM_AUTHTOK) available, the pam_sm_acct_mgmt(3PAM) function attempts to retrieve the user's account status without authenticating to the LDAP server as the user logging in. This procedure will succeed only if the LDAP server is Sun Java System Directory server 5.2 patch 4 or newer. The following options can be passed to the LDAP service module:

debug syslog(3C) debugging information at LOG_DEBUG level.

nowarn Turn off warning messages.

These options are case sensitive, and the options must be used exactly as presented here.

LDAP Password Management Module

LDAP password management is no longer supported by pam_ldap. Use pam_authtok_store(5) instead of pam_ldap for password change. pam_authtok_store(5) handles both the local and LDAP accounts and updates the passwords in all the repositories configured by nsswitch.conf(4).

エラー

The authentication service returns the following error codes:

PAM_SUCCESS	The uthentication was successful.
PAM_MAXTRIES	The maximum number of authentication attempts was exceeded.
PAM_AUTH_ERR	The authentication failed.
PAM_USER_UNKNOWN	No account is present for the user.
PAM_BUF_ERR	A memory buffer error occurred.
PAM_SYSTEM_ERR	A system error occurred.
PAM_IGNORE	The user's account was inactivated.

The account management service returns the following error codes:

PAM_SUCCESS	The user was allowed access to the account.
PAM_NEW_AUTHTOK_REQD	A new authentication token is required.
PAM_ACCT_EXPIRED	The user account has expired.
PAM_PERM_DENIED	The user was denied access to the account at this time.
PAM_USER_UNKNOWN	No account is present for the user.
PAM_BUF_ERROR	A memory buffer error occurred.
PAM_SYSTEM_ERR	A system error occurred.

使用例

例 1 Using pam_ldap With Authentication

The following is a configuration for the login service when using pam_ldap. The service name login can be substituted for any other authentication service such as dtlogin or su. Lines that begin with the # symbol are comments and are ignored.

```
# Authentication management for login service is stacked.
# If pam_unix_auth succeeds, pam_ldap is not invoked.
# The control flag "binding" provides a local overriding
# remote (LDAP) control. The "server_policy" option is used
# to tell pam_unix_auth.so.1 to ignore the LDAP users.

login   auth requisite  pam_authtok_get.so.1
login   auth required   pam_dhkeys.so.1
login   auth required   pam_unix_cred.so.1
login   auth binding    pam_unix_auth.so.1 server_policy
login   auth required   pam_ldap.so.1
```

例 2 Using pam_ldap With Account Management

The following is a configuration for account management when using pam_ldap. Lines that begin with the # symbol are comments and are ignored.

```
# Account management for all services is stacked
# If pam_unix_account succeeds, pam_ldap is not invoked.
# The control flag "binding" provides a local overriding
# remote (LDAP) control. The "server_policy" option is used
# to tell pam_unix_account.so.1 to ignore the LDAP users.

other   account  requisite      pam_roles.so.1
other   account  binding        pam_unix_account.so.1 server_policy
other   account  required       pam_ldap.so.1
```

例 3 Using pam_authtok_store With Password Management For Both Local and LDAP Accounts

The following is a configuration for password management when using pam_authtok_store. Lines that begin with the # symbol are comments and are ignored.

```
# Password management (authentication)
# The control flag "binding" provides a local overriding
# remote (LDAP) control. The server_policy option is used
# to tell pam_passwd_auth.so.1 to ignore the LDAP users.

passwd  auth binding  pam_passwd_auth.so.1 server_policy
passwd  auth required pam_ldap.so.1

# Password management (updates)
# This updates passwords stored both in the local /etc
# files and in the LDAP directory. The "server_policy"
# option is used to tell pam_authtok_store to
```

例 3 Using pam_authtok_store With Password Management For Both Local and LDAP Accounts *(続き)*

```
# follow the LDAP server's policy when updating
# passwords stored in the LDAP directory

other password required   pam_dhkeys.so.1
other password requisite  pam_authtok_get.so.1
other password requisite  pam_authtok_check.so.1
other password required   pam_authtok_store.so.1 server_policy
```

ファイル

`/var/ldap/ldap_client_file` `/var/ldap/ldap_client_cred`	The LDAP configuration files of the client. Do not manually modify these files, as these files might not be human readable. Use `ldapclient(1M)` to update these files.
`/etc/pam.conf`	PAM configuration file.
`/etc/pam.d/`*service*	Alternate PAM configuration files.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	MT-Safe with exceptions

関連項目

`idsconfig(1M)`, `ldap_cachemgr(1M)`, `ldapclient(1M)`, `libpam(3LIB)`, `pam(3PAM)`, `pam_sm_acct_mgmt(3PAM)`, `pam_sm_authenticate(3PAM)`, `pam_sm_chauthtok(3PAM)`, `pam_sm_close_session(3PAM)`, `pam_sm_open_session(3PAM)`, `pam_sm_setcred(3PAM)`, `syslog(3C)`, `pam.conf(4)`, `attributes(5)`, `ldap(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`, `pam_unix_auth(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own `PAM` handle.

The previously supported `use_first_pass` and `try_first_pass` options are obsolete in this version, are no longer needed, can safely be removed from `pam.conf(4)`, and are silently ignored. They might be removed in a future release. Password prompting must be provided for by stacking `pam_authtok_get(5)` before `pam_ldap` in the `auth` and `password` module stacks and `pam_passwd_auth(5)` in the `passwd` service `auth` stack (as described in the EXAMPLES section). The previously supported password update function is replaced in this release by the previously recommended use of `pam_authtok_store` with the `server_policy` option (as described in the EXAMPLES section).

The functions: pam_sm_setcred(3PAM), pam_sm_chauthtok(3PAM), pam_sm_open_session(3PAM), and pam_sm_close_session(3PAM) do nothing and return PAM_IGNORE in pam_ldap.

名前

pam_list – PAM account management module for UNIX

形式

```
pam_list.so.1
```

機能説明

The pam_list module implements pam_sm_acct_mgmt(3PAM), which provides functionality to the PAM account management stack. The module provides functions to validate that the user's account is valid on this host based on a list of users and/or netgroups in the given file. The users and netgroups are separated by newline character. Netgroups are specified with character '@' as prefix before name of netgroup in the list. The maximum line length is 1023 characters.

The username is the value of PAM_USER. The host is the value of PAM_RHOST or, if PAM_RHOST is not set, the value of the localhost as returned by gethostname(3C) is used.

If neither of the allow, deny, or compat options are specified, the module will look for +/- entries in the local /etc/passwd file. If this style is used, nsswitch.conf(4) must not be configured with compat for the passwd database. If no relevant +/- entry exists for the user, pam_list is not participating in result.

If compat option is specified then the module will look for +/- entries in the local /etc/passwd file. Other entries in this file will be counted as + entries. If no relevant entry exits for the user, pam_list will deny the access.

The following options can be passed to the module:

allow=	The full pathname to a file of allowed users and/or netgroups. Only one of allow= or deny= can be specified.
compat	Activate compat mode.
deny=	The full pathname to a file of denied users and/or netgroups. Only one of deny= or allow= can be specified.
debug	Provide syslog(3C) debugging information at the LOG_AUTH | LOG_DEBUG level.
user	The module should only perform netgroup matches on the username. This is the default option.
nouser	The username should not be used in the netgroup match.
host	Only the host should be used in netgroup matches.
nohost	The hostname should not be used in netgroup matches.
norole	Return PAM_IGNORE if the account (PAM_USER) is a role. This is the default.
role	Evaluate the rules even if PAM_USER is a role account.
user_host_exact	The user and hostname must be in the same netgroup.

エラー

The following error values are returned:

PAM_SERVICE_ERR	An invalid set of module options was specified in the PAM configuration (see pam.conf(4)) for this module, or the user/netgroup file could not be opened.
PAM_BUF_ERR	A memory buffer error occurred.
PAM_IGNORE	The module is ignored, as it is not participating in the result.
PAM_PERM_DENIED	The user is not on the allow list or is on the deny list.
PAM_SUCCESS	The account is valid for use at this time.
PAM_USER_UNKNOWN	No account is present for the user

使用例

例 1 Using pam_list in default mode

The changes to /etc/pam.conf would be:

```
other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account required        pam_list.so.1
```

The equivalent PAM configuration in /etc/pam.d/ would be the following entries in /etc/pam.d/other:

```
account requisite       pam_roles.so.1
account required        pam_unix_account.so.1
account required        pam_list.so.1
```

In the case of default mode or compat mode, the important lines in /etc/passwd appear as follows:

```
+loginname    - user is approved
-loginname    - user is disapproved
+@netgroup    - netgroup members are approved
-@netgroup    - netgroup members are disapproved
```

例 2 Using pam_list with allow file

The changes to /etc/pam.conf would be:

```
other   account requisite       pam_roles.so.1
other   account required        pam_unix_account.so.1
other   account required        pam_list.so.1 allow=/etc/users.allow
```

The equivalent PAM configuration in /etc/pam.d/ would be the following entries in /etc/pam.d/other:

```
account requisite       pam_roles.so.1
account required        pam_unix_account.so.1
```

例 2 Using pam_list with allow file (続き)

```
account required        pam_list.so.1 allow=/etc/users.allow
```

`/etc/users.allow` contains:

```
root
localloginname
remoteloginname
@netgroup
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	MT-Safe with exceptions

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multithreaded application uses its own PAM handle.

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `pam_sm_acct_mgmt(3PAM)`, `syslog(3C)`, `libpam(3LIB)`, `nsswitch.conf(4)`, `pam.conf(4)`, `attributes(5)`

名前

pam_passwd_auth – authentication module for password

形式

`pam_passwd_auth.so.1`

機能説明

`pam_passwd_auth` provides authentication functionality to the password service as implemented by `passwd(1)`. It differs from the standard PAM authentication modules in its prompting behavior. It should be the first module on the password service authentication stack.

The name of the user whose password attributes are to be updated must be present in the `PAM_USER` item. This can be accomplished due to a previous call to `pam_start(3PAM)`, or explicitly set by `pam_set_item(3PAM)`. Based on the current user-id and the repository that is to by updated, the module determines whether a password is necessary for a successful update of the password repository, and if so, which password is required.

The following options can be passed to the module:

`debug`	`syslog(3C)` debugging information at the `LOG_DEBUG` level
`nowarn`	Turn off warning messages
`server_policy`	If the account authority for the user, as specified by `PAM_USER`, is a server, do not apply the Unix policy from the passwd entry in the name service switch.

エラー

The following error codes are returned:

`PAM_BUF_ERR`	Memory buffer error
`PAM_IGNORE`	Ignore module, not participating in result
`PAM_SUCCESS`	Successfully obtains authentication token
`PAM_SYSTEM_ERR`	System error

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`passwd(1)`, `pam(3PAM)`, `pam_authenticate(3PAM)`, `pam_start(3PAM)`, `pam_set_item(3PAM)`, `syslog(3C)`, `libpam(3LIB)`, `pam.conf(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_unix_account(5)`, `pam_unix_auth(5)`, `pam_unix_session(5)`

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

This module relies on the value of the current real UID, this module is only safe for MT-applications that don't change UIDs during the call to pam_authenticate(3PAM).

名前

pam_pkcs11 – PAM Authentication Module for the PKCS#11 token libraries

形式

`pam_pkcs11.so [debug] [config_file=`*filename*`]`

機能説明

The `pam_pkcs11` module implements `pam_sm_authenticate(3PAM)`, which provides functionality to the PAM authentication stack. This module allows a user to login a system, using a X.509 certificate and its dedicated private key stored in a PKCS#11 token. This module currently supports the RSA algorithm only.

To verify the dedicated private key is truly associated with the X.509 certificate, the following verification procedure is performed in this module by default:

- Generate 128 random byte data
- Sign the random data with the private key and get a signature. This step is done in the PKCS#11 token.
- Verify the signature using the public key extracted from the certificate.

For the verification of the users' certificates, locally stored CA certificates as well as either online or locally accessible CRLs are used.

PAM CONFIGURATION

The `pam_pkcs11.so` service module can be used in the <auth> PAM chain. The program that needs a PAM service should be configured in `/etc/pam.conf` or `/etc/pam.d/`*service*. For details on how to configure PAM services, see `pam.conf(4)`.

The following example uses only `pam_pkcs11` for authentication:

```
login auth requisite pam_pkcs11.so.1
login autho required pam_unix_cred.so.1
```

The following example uses `pam_pkcs11` for authentication with fallback to standard UNIX authentication:

```
login auth sufficient pam_pkcs11.so.1
login auth requisite  pam_authtok_get.so.1
login auth required   pam_dhkeys.so.1
login auth required   pam_unix_cred.so.1
login auth required   pam_unix_auth.so.1
```

PAM_PKCS11 CONFIGURATION

To configure the `pam_pkcs11` module, you must have the following information:

- Which PKCS #11 token you are going to use
- Which mapper(s) you need, and if needed, how to create and edit the related mapping files
- The root Certificate Authority files, and if required, the Certificate Revocation Lists files
- The list of authorized users to login, and their corresponding certificates

To configure the `pam_pkcs11` module, you need to modify the `pam_pkcs11.conf` configuration file which is in the `/etc/security/pam_pkcs11` directory by default. For

detailed information on how to configure the pam_pkcs11 module, see the PAM-PKCS11 User Manual, available at the http://www.opensc-project.org/ web site, under the PAM PKCS#11 link.

The following example illustrates how to configure the pam_pkcs11 module for a user whose certificate and private key are stored in the Solaris pkcs11_softtoken keystore. This example uses the default certificate verification policy.

- Set up the PKCS#11 module.

  On Solaris, the PKCS#11 module should be set to /usr/lib/libpkcs11.so.1, the PKCS#11 Cryptographic Framework library.

- Set up the slot_description entry.

  Specifies the slot to be used. For example, slot_description = "Sun Crypto Softtoken". The default value for this entry is none which means to use the first slot with an available token.

  An administrator can use the cryptoadm list -vcommand to find all the available slots and their slot descriptions. For more information, see libpkcs11(3LIB) and cryptoadm(1M).

- Install or create user certificates and its dedicated private keys in the specific PKCS#11 token.

- Set up the certificate verification policy (cert_policy). If needed, set up CA certificate and CRL files.

  The certificate verification policy includes:

none	Perform no verification
ca	Perform CA check
signature	Perform a signature check to ensure that private and public key matches
crl_*xxx*	Perform various certificate revocation checking

  As this example uses the default policy, cert_policy = ca,signature, an administer needs to set up the CA certificates.

  – Copy the CA certificate to the /etc/security/pam_pkcs11/cacerts directory.

    A certificate that is self-signed is its own CA certificate. Therefore, in this example, the certificate is placed both in the Softtoken keystore and in the CA certificate directory.

  – Make hash links for CA certificates

    ```
    $ /etc/security/pam_pkcs11/make_hash_link.sh \
          /etc/security/pam_pkcs11/cacerts
    ```

- Set up the mappers and mapfiles.

When a X509 certificate is provided, there are no direct ways to map a certificate to a login. The `pam_pkcs11` module provides a configurable way with mappers to specify `cert-to-user` mapping.

Many mappers are provided by the `pam_pkcs11` module, for example, the common name (CN) mapper, the digest mapper, the Email mapper, or the LDAP mapper.

A user can configure a mapper list in the `pam_pkcs11.conf` file. The mappers in the list are used sequentially until the certificate is successfully matched with the user.

The default mapper list is as follows:

```
use_mappers = digest, cn, pwent, uid, mail, subject, null;
```

Some mappers do not require the specification of a mapfile, for example, the common name mapper. Other mappers require mapfiles, for example, the digest mapper. Some sample mapping files can be found in the `/etc/security/pam_pkcs11` directory.

オプション

The following options are supported:

`config_file=`*filename*
: Specify the configuration file. The default value is `/etc/security/pam_pkcs11/pam_pkcs11.conf`.

`debug`
: Enable debugging output.

ファイル

`/usr/lib/security/pam_pkcs11.so`
: `pam_pkcs11` module

`/usr/lib/pam_pkcs11/ldap_mapper.so`
: Mapper module.

`/usr/lib/pam_pkcs11/opensc_mapper.so`
: Mapper module.

`/usr/lib/pam_pkcs11/openssh_mapper.so`
: Mapper module.

`/etc/security/pam_pkcs11/pam_pkcs11.conf`
: Configuration file.

`/etc/security/pam_pkcs11/cacerts`
: Configuration directory. Stores the CA certificates.

`/etc/security/pam_pkcs11/crls`
: Configuration directory. Stores the CRL files.

`/etc/security/pam_pkcs11/digest_mapping.example`
: Sample mapfile.

`/etc/security/pam_pkcs11/subject_mapping.example`
: Sample mapfile.

`/etc/security/pam_pkcs11/mail_mapping.example`
Sample mapfile.

`/etc/security/pam_pkcs11/make_hash_link.sh`
Sample script.

Authors

`PAM-pkcs11` was originally written by MarioStrasser , `mast@gmx.net`.

Newer versions are from Juan Antonio Martinez, `jonsito@teleline.es`

属性

See `attributes(5)` for a description of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	library/security/pam/module/pam-pkcs11, SUNWpampkcs11r, SUNWpampkcs11-docs
Interface Stability	Uncommitted

関連項目

`pkcs11_inspect(1)`, `pklogin_finder(1)`, `cryptoadm(1M)`, `libpkcs11(3LIB)libpkcs11(3LIB)pam_sm_authenticate(3PAM)`, `pam.conf(4)`, `attributes(5)`, `pkcs11_softtoken(5)`

PAM-PKCS11 User Manual, available at the `http://www.opensc-project.org/` web site, under the `PAM PKCS#11` link.

名前

pam_rhosts_auth – authentication management PAM module using ruserok()

形式

`/usr/lib/security/pam_rhosts_auth.so.1`

機能説明

The rhosts PAM module, `/usr/lib/security/pam_rhosts_auth.so.1`, authenticates a user via the `rlogin` authentication protocol. Only `pam_sm_authenticate()` is implemented within this module. `pam_sm_authenticate()` uses the `ruserok(3SOCKET)` library function to authenticate the `rlogin` or `rsh` user. `pam_sm_setcred()` is a null function.

`/usr/lib/security/pam_rhosts_auth.so.1` is designed to be stacked on top of the `/usr/lib/security/pam_unix.so.1` module for both the `rlogin` and `rsh` services. This module is normally configured as *sufficient* so that subsequent authentication is performed only on failure of `pam_sm_authenticate()`. The following option may be passed in to this service module:

`debug` `syslog(3C)` debugging information at LOG_DEBUG level.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `ruserok(3SOCKET)`, `syslog(3C)`, `libpam(3LIB)`, `pam.conf(4)`, `attributes(5)`

注意事項

The interfaces in `libpam()` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_roles – Solaris Roles account management module

形式

`pam_roles.so.1`

機能説明

The `pam_roles` module implements `pam_sm_acct_mgmt(3PAM)`. It provides functionality to verify that a user is authorized to assume a role. It also prevents direct logins to a role. The `user_attr(4)` database is used to determine which users can assume which roles.

The PAM items `PAM_USER` and `PAM_AUSER`, and `PAM_RHOST` are used to determine the outcome of this module. `PAM_USER` represents the new identity being verified. `PAM_AUSER`, if set, represents the user asserting a new identity. If `PAM_AUSER` is not set, the real user ID of the calling service implies that the user is asserting a new identity. Notice that root can never have roles.

This module is generally stacked above the `pam_unix_account(5)` module.

The following options are interpreted:

`allow_remote` Allows a remote service to specify the user to enter as a role.

`debug` Provides `syslog(3C)` debugging information at the `LOG_DEBUG` level.

エラー

The following values are returned:

`PAM_IGNORE` If the type of the new user identity (`PAM_USER`) is "`normal`". Or, if the type of the new user identity is "`role`" and the user asserting the new identity (`PAM_AUSER`) has the new identity name in its list of roles.

`PAM_USER_UNKNOWN` No account is present for user.

`PAM_PERM_DENIED` If the type of the new user identity (`PAM_USER`) is "`role`" and the user asserting the new identity (`PAM_AUSER`) does not have the new identity name in its list of roles.

使用例

例 1 Using the pam_roles.so.1 Module

The following example is a `pam.conf(4)` fragment that demonstrates the use of the `pam_roles.so.1` module:

```
cron account required pam_unix_account.so.1

other account requisite pam_roles.so.1
other account required pam_unix_account.so.1
```

The equivalent configuration in `/etc/pam.d/` would be the following entry in `/etc/pam.d/cron`:

```
account required pam_unix_account.so.1
```

and the following entries in `/etc/pam.d/other`:

例 1 Using the pam_roles.so.1 Module (続き)

```
account requisite pam_roles.so.1
account required pam_unix_account.so.1
```

The cron service does not invoke pam_roles.so.1. Delayed jobs are independent of role assumption. All other services verify that roles cannot directly login. The "su" service (covered by the "other" service entry) verifies that if the new user is a role, the calling user is authorized for that role.

例 2 Allowing Remote Roles

Remote roles should only be allowed from remote services that can be trusted to provide an accurate PAM_AUSER name. This trust is a function of the protocol (such as sshd-hostbased).

The following example is a pam.conf(4) fragment that demonstrates the use of pam_roles configuration for remote roles for the sshd-hostbased service.

```
sshd-hostbased account requisite pam_roles.so.1 allow_remote
sshd-hostbased account required pam_unix_account
```

The equivalent configuration in /etc/pam.d/ would be the following entries in /etc/pam.d/sshd-hostbased:

```
account requisite pam_roles.so.1 allow_remote
account required pam_unix_account
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

roles(1), sshd(1M), su(1M), libpam(3LIB), pam(3PAM), pam_acct_mgmt(3PAM), pam_setcred(3PAM), pam_set_item(3PAM), pam_sm_acct_mgmt(3PAM), syslog(3C), pam.conf(4), user_attr(4), attributes(5), pam_authtok_check(5), pam_authtok_get(5), pam_authtok_store(5), pam_dhkeys(5), pam_passwd_auth(5), pam_unix_account(5), pam_unix_auth(5), pam_unix_session(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

This module should never be stacked alone. It never returns PAM_SUCCESS, as it never makes a positive decision.

The allow_remote option should only be specified for services that are trusted to correctly identify the remote user (that is, sshd-hostbased).

PAM_AUSER has replaced PAM_RUSER whose definition is limited to the rlogin/rsh untrusted remote user name. See pam_set_item(3PAM).

名前

pam_sample – a sample PAM module

形式

`/usr/lib/security/pam_sample.so.1`

機能説明

The SAMPLE service module for PAM is divided into four components: authentication, account management, password management, and session management. The sample module is a shared object that is dynamically loaded to provide the necessary functionality.

Sample Authentication Component

The SAMPLE authentication module provides functions to test the PAM framework functionality using the `pam_sm_authenticate(3PAM)` call. The SAMPLE module implementation of the `pam_sm_authenticate(3PAM)` function compares the user entered password with the password specified in the PAM configuration (see `pam.conf(4)`), or the string `test` if a default test password has not been set. The following options can be passed in to the SAMPLE Authentication module:

`debug`	Syslog debugging information at the `LOG_DEBUG` level.
`pass=newone`	Sets the password to be `newone`.
`first_pass_good`	The first password is always good when used with the use_first_pass or try_first_pass option.
`first_pass_bad`	The first password is always bad when used with the use_first_pass or try_first_pass option.
`always_fail`	Always returns `PAM_AUTH_ERR`.
`always_succeed`	Always returns `PAM_SUCCESS`.
`always_ignore`	Always returns `PAM_IGNORE`.
`use_first_pass`	Use the user's initial password (entered when the user is authenticated to the first authentication module in the stack) to authenticate with the SAMPLE module. If the passwords do not match, or if this is the first authentication module in the stack, quit and do not prompt the user for a password. It is recommended that this option only be used if the SAMPLE authentication module is designated as *optional* in the PAM configuration (see `pam.conf(4)`).
`try_first_pass`	Use the user's initial password (entered when the user is authenticated to the first authentication module in the stack) to authenticate with the SAMPLE module. If the passwords do not match, or if this is the first authentication module in the stack, prompt the user for a password. The SAMPLE module `pam_sm_setcred(3PAM)` function always returns `PAM_SUCCESS`.

Sample Account Management Component

The SAMPLE Account Management Component implements a simple access control scheme that limits machine access to a list of authorized users. The list of authorized users is supplied

as option arguments to the entry for the SAMPLE account management PAM module in the PAM configuration (see pam.conf(4)). Note that the module always permits access to the root super user.

The option field syntax to limit access is shown below: allow= *name[,name]* allow= *name [allow=name]*

The example pam.conf show below permits only larry to login directly. rlogin is allowed only for don and larry. Once a user is logged in, the user can use su if the user are sam or eric.

login	account	require	pam_sample.so.1	allow=larry
gdm	account	require	pam_sample.so.1	allow=larry
rlogin	account	require	pam_sample.so.1	allow=don allow=larry
su	account	require	pam_sample.so.1	allow=sam,eric

The debug and nowarn options are also supported.

Sample Password Management Component

The SAMPLE Password Management Component function (pam_sm_chauthtok(3PAM)), always returns PAM_SUCCESS.

Sample Session Management Component

The SAMPLE Session Management Component functions (pam_sm_open_session(3PAM), pam_sm_close_session(3PAM)) always return PAM_SUCCESS.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT Level	MT-Safe with exceptions

関連項目

pam(3PAM), pam_sm_authenticate(3PAM), pam_sm_chauthtok(3PAM), pam_sm_close_session(3PAM), pam_sm_open_session(3PAM), pam_sm_setcred(3PAM), libpam(3LIB), pam.conf(4), attributes(5)

警告

This module should never be used outside of a closed debug environment. The examples of the use_first_pass and try_first_pass options are obsolete for all other Solaris delivered PAM service modules

注意事項

The interfaces in libpam() are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_smbfs_login – PAM user credential authentication module for SMB/CIFS client login

形式

`pam_smb_cred.so.1`

機能説明

The `pam_smbfs_login` module implements `pam_sm_setcred(3PAM)` to provide functions that act equivalently to the `smbadm(1M)` `add-key` command.

This optional functionality is meant to be used only in environments that do not run Active Directory or Kerberos, but which synchronize passwords between Solaris clients and their CIFS/SMB servers.

This module permits the login password to be stored as if the `smbadm(1M)` `add-key` command was used to store a password for `PAM_USER` in the user or system default domain.

To use this functionality, add the following line to the `/etc/pam.d/login` file:

```
auth optional    pam_smbfs_login.so.1
```

Authentication service modules must implement both `pam_sm_authenticate(3PAM)` and `pam_sm_setcred(3PAM)`. In this module, `pam_sm_authenticate(3PAM)` always returns `PAM_IGNORE`.

The `pam_sm_setcred(3PAM)` function accepts the following flags:

`PAM_REFRESH_CRED`
Returns PAM_IGNORE.

`PAM_SILENT`
Suppresses messages.

`PAM_ESTABLISH_CRED`
`PAM_REINITIALIZE_CRED`
Stores the authentication token for PAM_USER in the same manner as the `smbadm(1M)` `add-key` command.

`PAM_DELETE_CRED`
Deletes the stored password for PAM_USER in the same manner as the `smbadm(1M)` `remove-key` command.

The following options can be passed to the `pam_smbfs_login` module:

`debug`
Produces `syslog(3C)` debugging information at the `LOG_AUTH` or `LOG_DEBUG` level.

`nowarn`
Suppresses warning messages.

エラー

Upon successful completion of `pam_sm_setcred(3PAM)`, `PAM_SUCCESS` is returned. The following error codes are returned upon error:

`PAM_USER_UNKNOWN`
User is unknown.

PAM_AUTHTOK_ERR
Password is bad.

PAM_AUTH_ERR
Domain is bad.

PAM_SYSTEM_ERR
System error.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

smbadm(1M), syslog(3C), libpam(3LIB), pam(3PAM), pam_setcred(3PAM), pam_sm(3PAM), pam_sm_authenticate(3PAM), pam_sm_chauthtok(3PAM), pam_sm_setcred(3PAM), pam.conf(4), attributes(5), smbfs(7FS)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_smb_passwd – SMB password management module

形式

`pam_smb_passwd.so.1`

機能説明

The `pam_smb_passwd` module enhances the PAM password management stack. This functionality supports the changing or adding of SMB passwords for local Solaris users. The Solaris SMB server uses SMB passwords to authenticate connected Solaris users. This module includes the `pam_sm_chauthtok(3PAM)` function.

The `pam_sm_chauthtok()` function accepts the following flags:

`PAM_PRELIM_CHECK`
Always returns `PAM_IGNORE`.

`PAM_SILENT`
Suppresses messages.

`PAM_UPDATE_AUTHTOK`
Updates or creates a new SMB local LM/NTLM hash for the user that is specified in `PAM_USER` by using the authentication information found in `PAM_AUTHTOK`. The LM hash is only created if the `smbd/lmauth_level` property value of the `smb/server` service is set to 3 or less. `PAM_IGNORE` is returned if the user is not in the local `/etc/passwd` repository.

The following options can be passed to the `pam_smb_passwd` module:

`debug`
Produces `syslog(3C)` debugging information at the `LOG_AUTH` or `LOG_DEBUG` level.

`nowarn`
Suppresses warning messages.

ファイル

`/var/smb/smbpasswd`
Stores SMB passwords for Solaris users.

エラー

Upon successful completion of `pam_sm_chauthtok()`, `PAM_SUCCESS` is returned. The following error codes are returned upon error:

`PAM_AUTHTOK_ERR`
Authentication token manipulation error

`PAM_AUTHTOK_LOCK_BUSY`
SMB password file is locked

`PAM_PERM_DENIED`
Permissions are insufficient for accessing the SMB password file

`PAM_SYSTEM_ERR`
System error

`PAM_USER_UNKNOWN`
User is unknown

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`smbd`(1M), `syslog`(3C), `libpam`(3LIB), `pam`(3PAM), `pam_chauthtok`(3PAM), `pam_sm`(3PAM), `pam_sm_chauthtok`(3PAM), `pam.conf`(4), attributes(5)

注意事項

The interfaces in `libpam`(3LIB) are MT-Safe *only* if each thread within the multi-threaded application uses its own PAM handle.

The `pam_smb_passwd.so.1` module should be stacked following all password qualification modules in the PAM password stack.

名前

pam_tsol_account – Trusted Extensions 用 PAM アカウント管理モジュール

形式

`/usr/lib/security/pam_tsol_account.so.1`

機能説明

PAM 用の Solaris Trusted Extensions サービスモジュール `/usr/lib/security/pam_tsol_account.so.1` は、ラベルに関連したアカウント制限を検査します。`pam_tsol_account.so.1` モジュールは、必要な機能をオンデマンドで提供する、動的にロード可能な共有オブジェクトです。パスは、PAM 構成ファイルで指定されます。

`pam_tsol_account.so.1` には、アカウント管理を実行するための関数 `pam_sm_acct_mgmt()` が含まれます。この関数は、ユーザーに許可されるラベル範囲を検査します。許可されるラベル範囲のデフォルトは、`label_encodings(4)` ファイル内に設定されています。これらのデフォルトは、`user_attr(4)` データベース内のエントリで上書きできます。

デフォルトでは、このモジュールは、大域ゾーンに接続するリモートホストが CIPSO ホストタイプであることが必要です。このポリシーを無効にするには、次に示すように、`pam.conf(4)` のエントリのオプションとして `allow_unlabeled` キーワードを追加します。

```
other  account required    pam_tsol_account allow_unlabeled
```

`/etc/pam.d/` 内の同等の PAM 構成では、`/etc/pam.d/other` 内に次のエントリが含まれます:

```
account required          pam_tsol_account allow_unlabeled
```

オプション

このモジュールに渡すことができるオプションを、次に示します。

`allow_unlabeled` ラベル付けされていないテンプレートタイプのホストからのリモート接続を許可します。

`debug` デバッグ情報を `LOG_DEBUG` レベルで提供します。`syslog(3C)` を参照してください。

戻り値

次の値が返されます。

`PAM_SUCCESS` アカウントは、今回およびこのラベルでの使用で有効です。

`PAM_PERM_DENIED` 現在のプロセスラベルがユーザーのラベル範囲外にある、プロセスのラベル情報を使用できない、リモートホストタイプが無効である、のいずれかです。

Other values 通常の PAM 操作と一致するエラーコードを返します。エラー関連の戻り値の詳細は、`pam(3PAM)` のマニュアルページを参照してください。

属性

次の属性については、 attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実
MT レベル	例外付きで MT-安全

libpam(3LIB) のインタフェースは、マルチスレッドアプリケーション内部の各スレッドが独自の PAM ハンドルを使用する場合にのみ、「MT- 安全」です。

関連項目

keylogin(1), libpam(3LIB), pam(3PAM), pam_sm_acct_mgmt(3PAM), pam_start(3PAM), syslog(3C), label_encodings(4), pam.conf(4), user_attr(4), attributes(5)

『Oracle Solaris 11.1 の管理: セキュリティーサービス』の第 14 章「プラグイン可能認証モジュールの使用」

注意事項

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

名前

pam_tty_tickets – PAM authentication module

形式

```
pam_tty_tickets.so.1 [timeout=minutes] [sudo-compat] [debug]
```

機能説明

The `pam_tty_tickets` module provides a mechanism for checking a ticket that was created by a prior successful authentication. Tickets by default validity of 5 minutes.

The default ticket location includes both the `source` (`PAM_AUSER`) and `destination` (`PAM_USER`) as well as the `tty` (`PAM_TTY`) for which it is valid.

The module can be configured using the `sudo–compat` option to store the tickets in the same location as `sudo`, though use of `sudo` is not required to use this feature.

The `pam_sm_setcred()` function creates a ticket for the user in the tickets directory.

The `pam_sm_authenticate()` function checks the timestamp on the ticket is no older than the timeout value, if is then it returns `PAM_SUCCESS`. If it is older then the ticket is removed and the module returns `PAM_IGNORE`.

This module is intended to be placed in the `auth` stack with the sufficient control flag.

No messages are produced by this module using the PAM conversation function. Some messages are sent to syslog for error conditions as as well as messages at `LOG_INFO` for ticket validity checking

The following options can be passed to the module:

`debug`	Debugging information is sent to `syslog` `LOG_AUTH|LOG_DEBUG`.
sudo-compat	Location of the per user (per `tty`) tickets, matches the `sudo` location. When this option is set `PAM_USER` must be root other wise the module returns `PAM_IGNORE` and tickets are not read or created.
timeout	Validity time in minutes for a ticket. The default is 5 minutes.

使用例

例 1 Using the Default Settings

The following is an excerpt of a sample `pam.conf` configuration file that has per `tty` tickets with the default time out (5 minutes) for users authenticating with su(1M):

```
su auth required    pam_unix_cred.so.1
su auth sufficient  pam_tty_tickets.so.1
su auth requisite   pam_authtok_get.so.1
su auth required    pam_dhkeys.so.1
su auth required    pam_unix_auth.so.1
```

例 2 Changing the Default Settings

The following example changes the defaults so that tickets are valid for 10 minutes and uses the `sudo` location:

例 2 Changing the Default Settings (続き)

```
su auth required    pam_unix_cred.so.1
su auth sufficient  pam_tty_tickets.so.1 sudo-compat timeout=10
su auth requisite   pam_authtok_get.so.1
su auth required    pam_dhkeys.so.1
su auth required    pam_unix_auth.so.1
```

エラー

`PAM_SUCCESS` Ticket is valid

`PAM_IGNORE` All other cases

ファイル

`/system/volatile/tty_tickets/<PAM_AUSER>/<PAM_USER>/<PAM_TTY>`
Default ticket location.

`/system/volatile/sudo/<PAM_AUSER>/<PAM_TTY>`
When used `sudo-compat` is set this file has the same format as those created by `sudo`.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	See below.

The `syslog` messages are Volatile. The module name, module options, and ticket locations are Committed.

関連項目

`su(1M)`, `sudo`(1M), `pam(3PAM)`, `pam_sm_authenticate(3PAM)`, `pam_sm_setcred(3PAM)`, `attributes(5)`

名前

pam_unix_account – PAM account management module for UNIX

形式

`pam_unix_account.so.1`

機能説明

`pam_unix_account` module implements `pam_sm_acct_mgmt()`, which provides functionality to the PAM account management stack. The module provides functions to validate that the user's account is not locked or expired and that the user's password does not need to be changed. The module retrieves account information from the configured databases in `nsswitch.conf(4)`.

The following options can be passed to the module:

`debug`	`syslog(3C)` debugging information at the `LOG_DEBUG` level
`nowarn`	Turn off warning messages
`server_policy`	If the account authority for the user, as specified by `PAM_USER`, is a server, do not apply the Unix policy from the passwd entry in the name service switch.

エラー

The following values are returned:

`PAM_UNIX_ACCOUNT`	User account has expired
`PAM_AUTHTOK_EXPIRED`	Password expired and no longer usable
`PAM_BUF_ERR`	Memory buffer error
`PAM_IGNORE`	Ignore module, not participating in result
`PAM_NEW_AUTHTOK_REQD`	Obtain new authentication token from the user
`PAM_PERM_DENIED`	The account is locked or has been inactive for too long
`PAM_SERVICE_ERR`	Error in underlying service module
`PAM_SUCCESS`	The account is valid for use at this time
`PAM_USER_UNKNOWN`	No account is present for the user

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `syslog(3C)`, `libpam(3LIB)`, `pam.conf(4)`, `nsswitch.conf(4)`, `attributes(5)`

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

Attempts to validate locked accounts are logged via syslog(3C) to the LOG_AUTH facility with a LOG_NOTICE severity.

名前

pam_unix_auth – PAM authentication module for UNIX

形式

pam_unix_auth.so.1

機能説明

The pam_unix_auth module implements pam_sm_authenticate(), which provides functionality to the PAM authentication stack. It provides functions that use crypt(3C) to verify that the password contained in the PAM item PAM_AUTHTOK is the correct password for the user specified in the item PAM_USER.

If PAM_AUSER and PAM_USER are both specified and PAM_USER is a role, the user_attr(4) keyword roleauth is checked to determine if the password that is checked is for the role (PAM_USER) or the assuming user (PAM_AUSER). If PAM_REPOSITORY is specified, the user's pass word is fetched from that repository. Otherwise, the default nsswitch.conf(4) repository is searched for that user.

For accounts in the name services which support automatic account locking, the account can be configured to be automatically locked (see user_attr(4) and policy.conf(4)) after multiple failed login attempts. For accounts that are configured for automatic locking, if authentication failure is to be returned, the failed login counter is incremented upon each failure. If the number of successive failures equals or exceeds the configured value, the account is locked and PAM_MAXTRIES is returned. The files (see passwd(4) and shadow(4)) and ldap (when configured with enableShadowUpdate true, see ldapclient(1M)), repositories support automatic account locking. A successful authentication by this module clears the failed login counter and reports the number of failed attempts since the last successful authentication.

Authentication service modules must implement both pam_sm_authenticate() and pam_sm_setcred(). To allow the authentication portion of UNIX authentication to be replaced, pam_sm_setcred() in this module always returns PAM_IGNORE. This module should be stacked with pam_unix_cred(5) to ensure a successful return from pam_setcred(3PAM).

The following options can be passed to the module:

nowarn
: Turn off warning messages.

server_policy
: If the account authority for the user, as specified by PAM_USER, is a server, do not apply the UNIX policy from the passwd entry in the name service switch.

nolock
: Regardless of the automatic account locking setting for the account, do not lock the account, increment or clear the failed login count. The nolock option allows for exempting account locking on a per service basis.

エラー

The following error codes are returned from `pam_sm_authenticate()`:

`PAM_AUTH_ERR`
Authentication failure.

`PAM_BUF_ERR`
Memory buffer error.

`PAM_IGNORE`
Ignores module, not participating in result.

`PAM_MAXTRIES`
Maximum number of retries exceeded.

`PAM_PERM_DENIED`
Permission denied.

`PAM_SUCCESS`
Successfully obtains authentication token.

`PAM_SYSTEM_ERR`
System error.

`PAM_USER_UNKNOWN`
No account present for user.

The following error codes are returned from `pam_sm_setcred()`:

`PAM_IGNORE`
Ignores this module regardless of the control flag.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`login(1)`, `passwd(1)`, `ldapclient(1M)`, `useradd(1M)`, `usermod(1M)`, `roleadd(1M)`, `rolemod(1M)`, `crypt(3C)`, `libpam(3LIB)`, `pam(3PAM)`, `pam_authenticate(3PAM)`, `pam_setcred(3PAM)`, `syslog(3C)`, `pam.conf(4)`, `passwd(4)`, `policy.conf(4)`, `nsswitch.conf(4)`, `shadow(4)`, `user_attr(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`, `pam_unix_session(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

If the PAM_REPOSITORY *item_type* is set and a service module does not recognize the type, the service module does not process any information, and returns PAM_IGNORE. If the PAM_REPOSITORY *item_type* is not set, a service module performs its default action.

名前

pam_unix_cred – PAM user credential authentication module for UNIX

形式

`pam_unix_cred.so.1`

機能説明

The `pam_unix_cred` module implements `pam_sm_setcred(3PAM)`. It provides functions that establish user credential information. It is a module separate from the `pam_unix_auth(5)` module to allow replacement of the authentication functionality independently from the credential functionality.

The `pam_unix_cred` module must always be stacked along with whatever authentication module is used to ensure correct credential setting.

Authentication service modules must implement both `pam_sm_authenticate()` and `pam_sm_setcred()`.

`pam_sm_authenticate()` in this module always returns `PAM_IGNORE`.

`pam_sm_setcred()` initializes the user's project, privilege sets and initializes or updates the user's audit context if it hasn't already been initialized. The following flags may be set in the flags field:

`PAM_ESTABLISH_CRED`
`PAM_REFRESH_CRED`
`PAM_REINITIALIZE_CRED`
: Initializes the user's project to the project specified in `PAM_RESOURCE`, or if `PAM_RESOURCE` is not specified, to the user's default project. Establishes the user's privilege sets.

  If the audit context is not already initialized and auditing is configured, these flags cause the context to be initialized to that of the user specified in `PAM_AUSER` (if any) merged with the user specified in `PAM_USER` and host specified in `PAM_RHOST`. If `PAM_RHOST` is not specified, `PAM_TTY` specifies the local terminal name. Attributing audit to `PAM_AUSER` and merging `PAM_USER` is required for correctly attributing auditing when the system entry is performed by another user that can be identified as trustworthy.

  If the audit context is already initialized, the `PAM_REINITIALIZE_CRED` flag merges the current audit context with that of the user specified in `PAM_USER`. `PAM_REINITIALIZE_CRED` is useful when a user is assuming a new identity, as with `su(1M)`.

`PAM_DELETE_CRED`
: This flag has no effect and always returns `PAM_SUCCESS`.

The following options are interpreted:

`debug` Provides `syslog(3C)` debugging information at the `LOG_DEBUG` level.

`nowarn` Disables any warning messages.

エラー

Upon successful completion of `pam_sm_setcred()`, `PAM_SUCCESS` is returned. The following error codes are returned upon error:

`PAM_CRED_UNAVAIL`	Underlying authentication service cannot retrieve user credentials
`PAM_CRED_EXPIRED`	User credentials have expired
`PAM_USER_UNKNOWN`	User is unknown to the authentication service
`PAM_CRED_ERR`	Failure in setting user credentials
`PAM_BUF_ERR`	Memory buffer error
`PAM_SYSTEM_ERR`	System error

The following values are returned from `pam_sm_authenticate()`:

`PAM_IGNORE`	Ignores this module regardless of the control flag

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`ssh(1)`, `su(1M)`, `settaskid(2)`, `libpam(3LIB)`, `getprojent(3PROJECT)`, `pam(3PAM)`, `pam_set_item(3PAM)`, `pam_sm_authenticate(3PAM)`, `syslog(3C)`, `setproject(3PROJECT)`,`pam.conf(4)`, `nsswitch.conf(4)`, `project(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_auth(5)`, `pam_unix_account(5)`, `pam_unix_session(5)`, `privileges(5)`

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

If this module is replaced, the audit context and credential may not be correctly configured.

名前

pam_unix_session – session management PAM module for UNIX

形式

`pam_unix_session.so.1`

機能説明

The `pam_unix_session` module implements `pam_sm_open_session(3PAM)` and `pam_sm_close_session(3PAM)`.

`pam_sm_open_session()` updates the `/var/adm/lastlog` file with the information contained in the `PAM_USER`, `PAM_TTY`, and `PAM_RHOST` items. `pam_unix_account(5)` uses this account to determine the previous time the user logged in.

`pam_sm_close_session()` is a null function.

The following options can be passed to the module:

debug `syslog(3C)` debugging information at the `LOG_DEBUG` level

エラー

Upon successful completion, `PAM_SUCCESS` is returned. The following error codes are returned upon error:

`PAM_SESSION_ERR` Cannot make or remove the entry for the specified session (PAM_TTY is not present).

`PAM_USER_UNKNOWN` No account is present for *user*.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT Level	MT-Safe with exceptions

関連項目

`pam(3PAM)`, `pam_authenticate(3PAM)`, `syslog(3C)`, `libpam(3LIB)`, `pam.conf(4)`, `nsswitch.conf(4)`, `attributes(5)`, `pam_authtok_check(5)`, `pam_authtok_get(5)`, `pam_authtok_store(5)`, `pam_dhkeys(5)`, `pam_passwd_auth(5)`, `pam_unix_account(5)`,`pam_unix_auth(5)`,

注意事項

The interfaces in `libpam(3LIB)` are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

名前

pam_user_policy – PAM user authentication policy module

形式

`pam_user_policy.so.1`

機能説明

The `pam_user_policy` module causes a user-specific PAM configuration to be evaluated and returns the result of evaluating such a configuration.

The `pam_user_policy` module implements all PAM service module functions.

The PAM configuration to evaluate is determined by looking for a `pam_policy` key in a user's attributes (see `user_attr(4)`) or profiles (see `prof_attr(4)`), or failing that, by looking at the default profiles granted in `policy.conf(4)`. If no PAM configuration is found, the "unix" policy is used.

Failure to obtain a user name is considered an error (see below).

This module should generally be stacked as the first module, possibly as the only module, in a PAM service configuration using a *control_flag* value of 'sufficient' or 'binding', depending on the contents of the user-specific PAM configuration.

The pathname to the user-specific PAM configuration file passed to `pam_eval(3PAM)` must be absolute so `pam_user_policy` prepends "`/etc/security/pam_policy`" to any non-absolute PAM configuration pathnames.

The following option can be passed to the module:

`debug` `syslog(3C)` debugging information at the `LOG_DEBUG` level

Authentication Module

The `pam_get_user(3PAM)` function is used to retrieve the current user name and sets this to be the value of `PAM_USER` if `PAM_USER` was not already set. If no user name can be obtained, `PAM_USER_UNKNOWN` is returned.

The `pam_user_policy` authentication module then looks up the name of a PAM configuration file to use for that user as described above and evaluates the named configuration by calling `pam_eval()` with the same flags as were passed to the `pam_user_policy` authentication module.

The configuration file name found or the default "unix" is saved as module data (see `pam_set_data(3PAM)` for use by other `pam_user_policy` modules.

Other Modules

If the `PAM_USER` item is not set to a non-empty string then `pam_user_policy` returns `PAM_USER_UNKNOWN` immediately. If a PAM configuration file name was saved as module data by a previous call to a `pam_user_policy` module, then that configuration will be used; otherwise a PAM configuration will be looked up as described above. The service module then evaluates the named configuration by calling `pam_eval()` with the same flags as were passed to the service module.

The configuration file name found or the default "unix" is saved as module data (see `pam_set_data(3PAM)`) for use by other `pam_user_policy` modules.

戻り値

If PAM_USER is not set or cannot be obtained, the module's service functions return PAM_USER_UNKNOWN. If module-specific data cannot be stored, PAM_SERVICE_ERR is returned. Failure to allocate resources causes the module to return PAM_BUF_ERR. Otherwise the value returned by pam_eval() is returned.

ファイル

A number of pam.conf files for inclusion by pam_user_policy can be found in /etc/security/pam_policy:

unix	Use only Unix passwords for authentication, account management, and password management.
krb5_only	Use Kerberos V5 only for authentication, account management, and password management.
krb5_first	Use Kerberos V5 for authentication with fallback on Unix authentication, use Kerberos V5 for account management and password management for Kerberos users and Unix for account management and password management for Unix users.
krb5_optional	Use Unix for authentication, account management, and password management and then optionally using Kerberos V5 for authentication, account management and password management for Kerberos users.
ldap	Use pam_ldap(5) for authentication, account management, and password management for LDAP users and Unix for authentication, account management, and password management for Unix users.
any	Try Kerberos V, LDAP and Unix, in that order, and as sufficient, for authentication, account management, and password management.

使用例

例 1 Authenticate a user with Kerberos V5 for all PAM services.

In the following example, user 'larry' should only be authenticated with Kerberos V5 for all PAM services.

```
$ usermod -K pam_policy=krb5_only larry
```

例 2 Use the PAM configuration /etc/security/pam_policy/custom for a user.

In the following example, the PAM configuration /etc/security/pam_policy/custom should be used for user 'curly'. This custom PAM configuration might have different configurations for different PAM services, such as requiring Unix authentication for console logins but Kerberos V5 for all other PAM services.

```
$ usermod -K pam_policy=custom curly
```

例 3 Create a new profile.

The following example creates a new profile named “PAM Per-User Policy of LDAP” and assign it to user 'moe' indicating that pam_ldap(5) should be used for all PAM services. Alternatively the profile could be assigned to all users by adding it to PROFS_GRANTED in policy.conf(4).

```
$ profiles -p "PAM Per-User Policy of LDAP" \
   'set desc="Profile which sets pam_policy=ldap";
   set pam_policy=ldap; exit;'
$ usermod -P "PAM Per-User Policy of LDAP" moe
```

例 4 Add a new user.

The following example adds a new user named 'shemp' who uses the PAM configuration /usr/local/etc/pam.conf for all PAM services.

```
$ useradd -K pam_policy=/usr/local/etc/pam.conf shemp
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	MT-Safe with exceptions

関連項目

libpam(3LIB), pam(3PAM), pam_eval(3PAM), pam_get_user(3PAM), pam_set_data(3PAM), syslog(3C), pam.conf(4), policy.conf(4), prof_attr(4), user_attr(4), attributes(5), pam_ldap(5)

注意事項

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multithreaded application uses its own PAM handle.

名前

pam_zfs_key – PAM user credential module for ZFS

形式

```
pam_zfs_key.so.1 [create] [homes=]
```

機能説明

The pam_zfs_key module implements pam_sm_setcred(3PAM) and pam_sm_chauthtok(3PAM).

The pam_zfs_key module provides functions that allow loading and changing of the ZFS encryption passphrase for encrypted file systems that are mounted at the users home directory location. Authentication service modules must implement both pam_sm_authenticate() and pam_sm_setcred().

pam_sm_authenticate() in this module always returns PAM_IGNORE. If they are located at a different location then the module option, homes= can be used to specify that. It is the ZFS dataset name, not the mountpoint, which would usually be /export/home/.

The last component of the ZFS dataset name must match the value of PAM_USER, that is, the users login name. If the users home directory is a local ZFS filesystem with encryption enabled and the ZFS keysource property is set to passphrase,prompt, on pam_sm_setcred() this module manages the keys as follows:

PAM_DELETE_CRED
: Attempts to umount the filesystem and unload the key. This often fails because there are still processes with the user's home directory as the current working directory. A force module option is provided to attempt a forced unmount first.

PAM_ESTABLISH_CRED
: Attempts to use the value of PAM_AUTHTOK to load the key for the ZFS dataset and mount it.

  If PAM_AUTHTOK is not the correct passphrase, the user is prompted once for an alternate. This value is never stored in PAM_AUTHTOK, even if it is the correct passphrase for the ZFS dataset that is the users home directory. This is equivalent to:

  ```
  zfs key -l rpool/export/home/$USER
  ```

If no ZFS file system exists for the user and the create module option is provided, a new one is created. The ZFS encryption property defaults to on in this case unless the encryption= property is set for the module to override it.

If the create module option is not provided and no ZFS file system exists for the user, the module returns PAM_IGNORE.

The newly created ZFS file system has the following ZFS delegations specified for the user for which it is created: key,keychange,mount. In these cases keysource is always set to passphrase,prompt.

When pam_sm_chauthtok(3PAM) is called, for example, on password change, this module attempts to change the passphrase for the ZFS dataset to match the value in PAM_AUTHTOK. This is equivalent to running:

```
zfs key -c rpool/export/home/$USER
```

This requires that the user have the keychange delegation, as password change usually runs as the user.

The following mount options are supported:

create	Create new ZFS datasets
encryption	Set the ZFS encryption property for create
force	Attempt a umount2(2) with a MS_FORCE of the dataset when doing PAM_DELETE_CRED.
homes=	Alternate location of ZFS datasets for user home directories. The default is rpool/export/home.
nowarn	Do not provide any error messages or warnings.

使用例

例 1 Using pam_zfs_key in Default Mode

The following example uses pam_zfs_key in default mode.

```
gdm     auth requisite          pam_authtok_get.so.1
gdm     auth required           pam_dhkeys.so.1
gdm     auth required           pam_unix_cred.so.1
gdm     auth required           pam_unix_auth.so.1
gdm     auth optional           pam_zfs_key.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password required       pam_authtok_store.so.1
other   password optional       pam_zfs_key.so.1
```

例 2 Specifying an Alternate ZFS Dataset

The following example specifies an alternate ZFS dataset location for the home directory file systems. New entries should be created if they are not present using aes-256-gcm as the ZFS encryption property setting.

```
gdm auth requisite  pam_authtok_get.so.1
gdm auth required   pam_dhkeys.so.1
gdm auth required   pam_unix_cred.so.1
gdm auth required   pam_unix_auth.so.1
gdm auth optional   pam_zfs_key.so.1 homes=tank/users \
create encryption=aes-256-gcm
```

例 3 Making it Mandatory for the ZFS Dataset to Mount

The following example makes it mandatory for the ZFS dataset to mount and ensures the passphrase always stays in sync with the login password.

```
gdm     auth requisite          pam_authtok_get.so.1
gdm     auth required           pam_dhkeys.so.1
gdm     auth required           pam_unix_cred.so.1
gdm     auth required           pam_unix_auth.so.1
gdm     auth required           pam_zfs_key.so.1

other   password required       pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password requisite      pam_zfs_key.so.1
other   password required       pam_authtok_store.so.1
```

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
MT-Level	MT-Safe with exceptions. See below.

The interfaces in libpam(3LIB) are MT-Safe only if each thread within the multi-threaded application uses its own PAM handle.

関連項目

zfs(1M), umount2(2), pam.conf(4), libpam(3LIB), pam(3PAM), pam_sm_chauthtok(3PAM), pam_sm_setcred(3PAM), attributes(5)

名前

pkcs11_kernel – カーネル暗号化フレームワークへの PKCS#11 インタフェース

形式

```
/usr/lib/security/pkcs11_kernel.so
/usr/lib/security/64/pkcs11_kernel.so
```

機能説明

pkcs11_kernel.so オブジェクトは、非公開インタフェースを使用してカーネル暗号化フレームワークと通信することで、RSA Security Inc. の PKCS#11 暗号化トークンインタフェース (Cryptoki) v2.20 仕様を実装します。

一意のハードウェアプロバイダはそれぞれ 1 つの PKCS#11 スロットで表されます。カーネル暗号化フレームワークのハードウェアプロバイダを備えていないシステムでは、この PKCS#11 ライブラリにスロットは存在しません。

このライブラリによって提供される PKCS#11 メカニズムは、使用可能なハードウェアプロバイダによって決まります。

アプリケーション開発者は、pkcs11_kernel.so に直接リンクするのではなく、libpkcs11.so にリンクするようにしてください。libpkcs11(3LIB) を参照してください。

libpkcs11(3LIB) のリストに示されている標準の PKCS#11 関数は、次のものを除きすべて実装されています。

```
C_DecryptDigestUpdate
C_DecryptVerifyUpdate
C_DigestEncryptUpdate
C_GetOperationState
C_InitToken
C_InitPIN
C_SetOperationState
C_SignEncryptUpdate
C_WaitForSlotEvent
```

これらの関数を呼び出すと、CKR_FUNCTION_NOT_SUPPORTED が返されます。

バッファーは 2M バイトを超えることはできません。たとえば、C_Encrypt() は、平文用の 2M バイトのバッファーと暗号化テキスト用の 2M バイトのバッファーを使用すると呼び出すことができます。

C_FindObjects() の呼び出しで取得できるオブジェクトハンドルの最大数は 512 です。

暗号化処理に使用できるカーネルメモリーの最大量は、project.max-crypto-memory リソース制御によって制限されます。カーネル内のバッファーおよびセッション関連の構造体の割り当ては、このリソース制御に対してチャージされます。

戻り値

実装されている各関数の戻り値は、RSA PKCS#11 v2.20 仕様で定義されています。http://www.rsasecurity.com を参照してください。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実
MT レベル	例外付きで MT-安全。RSA PKCS#11 v2.20 のセクション 6.6.2 を参照してください。
標準	PKCS#11 v2.20

関連項目

cryptoadm(1M), rctladm(1M), libpkcs11(3LIB), attributes(5), pkcs11_softtoken(5)

RSA PKCS#11 v2.20 http://www.rsasecurity.com

注意事項

アプリケーションが PKCS#11 スロットに対してセッションを開いていると、対応するハードウェアプロバイダのドライバは読み込み解除できなくなります。ハードウェアプロバイダのドライバを読み込み解除できるようにするには、そのプロバイダに対して PKCS#11 セッションを開いているアプリケーションを管理者が閉じる必要があります。

名前

pkcs11_kms – Oracle Key Manager の RSA PKCS#11 プロバイダ

形式

```
/usr/lib/security/pkcs11_kms.so

/usr/lib/security/64/pkcs11_kms.so
```

機能説明

pkcs11_kms.so オブジェクトは、Oracle Key Manager (OKM) KMS エージェントプロトコルを使用して Oracle Key Manager アプライアンス (KMA) と対話するために、RSA Security Inc. PKCS#11 Cryptographic Token Interface (Cryptoki)、v2.20 仕様を実装します。このプロバイダは PKCS#11 仕様を実装し、(非公開) KMS クライアントプロトコルを使用してリモート OKM と通信します。

このプロバイダでサポートされている PKCS#11 メカニズムは、CKM_AES_KEY_GEN、CKM_AES_CBC_PAD、および CKM_AES_CBC です。

このプロバイダでサポートされている PKCS#11 インタフェースは次のとおりです。

```
C_Initialize
C_Finalize
C_GetInfo
C_GetAttributeValue
C_SetAttributeValue
C_GetFunctionList
C_GetSlotList
C_GetSlotInfo
C_GetTokenInfo
C_GetMechanismList
C_GetMechanismInfo
C_InitToken
C_SetPIN
C_Login
C_Logout
C_FindObjectsInit/C_FindObjects/C_FindObjectsFinal
C_GenerateKey
C_EncryptInit/C_Encrypt/C_EncryptFinal
C_DecryptInit/C_Decrypt/C_DecryptFinal
C_DestroyObject
C_OpenSession
C_CloseSession
C_CloseAllSessions
C_GetSessionInfo
C_CreateObject
C_CopyObject
C_GetObjectSize
C_EncryptUpdate
C_DecryptUpdate
```

その他の関数を呼び出すと、すべて CKR_FUNCTION_NOT_SUPPORTED が返されます。

前提条件

pkcs11_kms プロバイダは、OKM にアクセスできるシステムでのみ使用できます。OKM 管理者は、OKM にアクセスする各ユーザー (またはアプリケーション) について、エージェント ID を構成する必要があります。これには OKM 管理ツールに含まれている OKM ユーティリティーを使用します。これらのユーティリティーは Oracle Solaris には含まれていません。

OKM 管理者が KMA を使用できるように構成し、パラメータをクライアント (Oracle Solaris のユーザーまたはアプリケーション) に通知したあと、Oracle Solaris PKCS#11 KMS プロバイダを初期化して使用できるようにします。

KMS プロバイダの初期化には、kmscfg(1M) ユーティリティーを使用します。ローカルプロバイダの構成ファイルを使用できるように初期化するには、ユーザーは kmscfg に少なくともプロファイルの名前、OKM エージェント ID、プロファイルのセキュリティー保護に使用されている初期パスワード、および KMA の IP アドレスを入力する必要があります。詳細については、kmscfg(1M) のマニュアルページを参照してください。

kmscfg を実行し、ローカルのトークン名前空間を構成したあと、ユーザーはトークンを使用するために初期化できます。トークンを初期化するには、pktool(1) コマンドを次のように使用します。

```
$ pktool inittoken currlabel=KMS
```

ユーザーは KMS プロバイダを使用するために初期化する前に、SO (セキュリティー責任者) のデフォルトの PIN を入力する必要があります。デフォルトの SO PIN は、OKM 管理者が最初に OKM エージェントを設定したときに使用したものです。トークンの初期化を行うユーザーは、このパスフレーズを知らなければプロバイダを初期化できません。

プロバイダを初期化したあとは、ユーザー PIN をデフォルト値から変更できます。PIN の値を変更するには、再度 pktool(1) を使用します。

ローカル PIN を変更するには、次のコマンドを使用します。

```
$ pktool setpin token=KMS
```

pktool setpin 操作または C_Login() 関数と C_SetPIN() 関数に設定する PIN には、長さが 1 から 256 までで、空文字を含まない、任意の文字列を使用できます。

トークンへのアクセス

ユーザーは自分のトークンを初期化したあと、pktool(1)、decrypt(1)、encrypt(1) で、または PKCS11 アプリケーションを作成することにより、KMS トークンを指定してトークンの使用を開始できます。

使用例

例 1　Oracle Key Manager に鍵を作成する

次のコマンドは、Oracle Key Manager に鍵を作成します。

```
$ pktool genkey token=KMS label=mykey1 keytype=aes keylen=256
```

例 2　Oracle Key Manager の鍵を使用してファイルを暗号化する

次のコマンドは、Oracle Key Manager の鍵を使用してファイルを暗号化します。

```
$ encrypt -a aes -K mykey1 -T KMS -i input.txt -o output.enc
```

例 3　Oracle Key Manager の鍵を使用してファイルを復号化する

次のコマンドは、Oracle Key Manager の鍵を使用してファイルを復号化します。

```
$ decrypt -a aes -K mykey1 -T KMS -i output.enc -o output.txt
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	/system/library/security/crypto/pkcs11_kms
インタフェースの安定性	確実
MT レベル	例外付きで MT-安全。下記を参照。
標準	PKCS#11 v2.20

「MT-安全」属性の例外については、RSA PKCS#11 v2.20 のセクション 6.6.2 を参照してください。

関連項目

decrypt(1), encrypt(1), pktool(1), cryptoadm(1M), kmscfg(1M), libpkcs11(3LIB), attributes(5)

『KMS 2.2: Administration Guide』

『Oracle Key Manager (OKM) Administration Guide』

注意事項

pkcs11_kms.so は、専用のディレクトリを使用して、構成ファイルおよび KMA への接続を初期化するために必要なほかのデータを保持します。この専用ディレクトリは、それが最初に作成されたホストに対してローカルなディレクトリです。デフォルトでは、KMS トークンのディレクトリ領域は /var/user/$USERNAME/kms です。デフォルトの KMS ディレクトリは、kmscfg(1M)、decrypt(1)、encrypt(1)、および pktool(1) コマンドを使用する前に KMSTOKEN_DIR 環境変数を設定することによってオーバーライドできます。

PKCS#11 クライアントでは、Oracle Key Manager のソフトウェアバージョン 2.4 がインストールされていることが必要です。

PKCS#11 クライアントが、複数システムから同じエージェント ID を使用している場合、このエージェントはワンタイムパスフレーズフラグを設定しないで作成する必要があります。このオプションは、OKM クラスタの一部のメンバーが 2.4 より前の OKM ソフトウェアバージョンを実行している場合は使用できません。エージェントの作成の詳細は、『Oracle Key Manager (OKM) 管理ガイド』を参照してください。

OKM エージェントは、PKCS#11 クライアントの鍵の作成に使用される前に、デフォルト鍵グループが割り当てられる必要があります。デフォルト鍵グループがエージェントに割り当てられない場合、操作は `CKR_PIN_INCORRECT` エラーで失敗します。エージェントへの鍵グループ割り当ての詳細については、『Oracle Key Manager (OKM) Administration Guide』を参照してください。

名前

pkcs11_softtoken – ソフトウェアの RSA PKCS#11 ソフトトークン

形式

```
/usr/lib/security/pkcs11_softtoken.so
/usr/lib/security/64/pkcs11_softtoken.so
```

機能説明

pkcs11_softtoken.so オブジェクトは、RSA Security Inc. の PKCS#11 暗号化トークンインタフェース (Cryptoki) v2.20 仕様をソフトウェアに実装します。この PKCS#11 の実装により、トークンオブジェクトの永続的ストレージが提供されます。

アプリケーション開発者は、pkcs11_softtoken.so に直接リンクするのではなく、libpkcs11.so にリンクするようにしてください。libpkcs11(3LIB) を参照してください。

実装されている暗号化アルゴリズムは、DES、3DES、AES、Blowfish、RC4、MD5、SHA1、SHA224、SHA256、SHA384、SHA512、RSA および ECC です。

libpkcs11(3LIB) のリストに示されている標準の PKCS#11 関数は、次のものを除きすべて実装されています。

```
C_GetObjectSize
C_InitPIN
C_InitToken
C_WaitForSlotEvent
```

これらの関数を呼び出すと、CKR_FUNCTION_NOT_SUPPORTED が返されます。

サポートされている RSA PKCS#11 v2.20 メカニズムは次のとおりです。

```
CKM_RSA_PKCS_KEY_PAIR_GEN
CKM_RSA_PKCS
CKM_RSA_X_509

CKM_DSA_KEY_PAIR_GEN
CKM_DSA
CKM_DSA_SHA1

CKM_DH_PKCS_KEY_PAIR_GEN
CKM_DH_PKCS_DERIVE

CKM_EC_KEY_PAIR_GEN
CKM_ECDSA
CKM_ECDSA_SHA1
CKM_ECDH1_DERIVE

CKM_DES_KEY_GEN
CKM_DES_ECB
CKM_DES_CBC
CKM_DES_CBC_PAD
```

```
CKM_DES3_KEY_GEN
CKM_DES3_ECB
CKM_DES3_CBC
CKM_DES3_CBC_PAD

CKM_AES_KEY_GEN
CKM_AES_ECB
CKM_AES_CBC
CKM_AES_CBC_PAD
CKM_AES_CTR

CKM_BLOWFISH_KEY_GEN
CKM_BLOWFISH_CBC

CKM_RC4_KEY_GEN
CKM_RC4

CKM_MD5_RSA_PKCS
CKM_SHA1_RSA_PKCS
CKM_SHA224_RSA_PKCS
CKM_SHA256_RSA_PKCS
CKM_SHA384_RSA_PKCS
CKM_SHA512_RSA_PKCS

CKM_MD5
CKM_SHA_1
CKM_SHA224
CKM_SHA256
CKM_SHA384
CKM_SHA512

CKM_MD5_HMAC
CKM_MD5_HMAC_GENERAL
CKM_SHA_1_HMAC
CKM_SHA_1_HMAC_GENERAL
CKM_SHA224_HMAC
CKM_SHA256_HMAC
CKM_SHA224_HMAC_GENERAL
CKM_SHA256_HMAC_GENERAL
CKM_SHA384_HMAC
CKM_SHA384_HMAC_GENERAL

CKM_MD5_KEY_DERIVATION
CKM_SHA1_KEY_DERIVATION
CKM_SHA224_KEY_DERIVATION
CKM_SHA256_KEY_DERIVATION
CKM_SHA384_KEY_DERIVATION
CKM_SHA512_KEY_DERIVATION
```

CKM_SSL3_PRE_MASTER_KEY_GEN
CKM_SSL3_MASTER_KEY_DERIVE
CKM_SSL3_KEY_AND_MAC_DERIVE
CKM_SSL3_MASTER_KEY_DERIVE_DH
CKM_TLS_PRE_MASTER_KEY_GEN
CKM_TLS_MASTER_KEY_DERIVE
CKM_TLS_KEY_AND_MAC_DERIVE
CKM_TLS_MASTER_KEY_DERIVE_DH

次に示す各タイプの鍵オブジェクトには、トークン固有の属性がいくつかあります。これらは、オブジェクトの作成、鍵/鍵ペアの生成、および鍵の派生の結果として、デフォルトで true に設定されます。

公開鍵オブジェクト	CKA_ENCRYPT、CKA_VERIFY、 CKA_VERIFY_RECOVER
秘密鍵オブジェクト	CKA_DECRYPT、CKA_SIGN、 CKA_SIGN_RECOVER、CKA_EXTRACTABLE
秘密鍵オブジェクト	CKA_ENCRYPT、CKA_DECRYPT、 CKA_SIGN、CKA_VERIFY、CKA_EXTRACTABLE

サポートされている証明書オブジェクトは次のとおりです。

CKC_X_509	CKC_X_509 証明書オブジェクトでサポートされている属性は、CKA_SUBJECT、CKA_VALUE 、CKA_LABEL、CKA_ID、CKA_ISSUER 、CKA_SERIAL_NUMBER、および CKA_CERTIFICATE_TYPE です。
CKC_X_509_ATTR_CERT	CKC_X_509_ATTR_CERT 証明書オブジェクトでサポートされている属性 は、CKA_OWNER、CKA_VALUE、CKA_LABEL、CKA_SERIAL_NUMBER、CKA_AC_ISS 、CKA_ATTR_TYPES、および CKA_CERTIFICATE_TYPE です。

テンプレートに一致するオブジェクトの検索操作は C_FindObjectsInit で実行されます。一致したオブジェクトは、以降の C_FindObjects 操作のためにキャッシュされます。

pkcs11_softtoken.so オブジェクトにより、トークンオブジェクトを保存するためのファイルシステムベースの永続的なトークンオブジェクトストアが提供されます。トークンオブジェクトストアのデフォルトの場所は、getpwuid_r () から返されるユーザーのホームディレクトリです。ユーザーは ${SOFTTOKEN_DIR} 環境変数を使用すると、デフォルトの場所を変更できます。

トークンオブジェクトストアが一度も初期化されていない場合、C_Login () 関数は CKR_OK を返すことがありますが、ユーザーは非公開のトークンオブジェクトの作成、生成、派生、または検索を行うことはできず、CKR_PIN_EXPIRED が返されます。

ユーザーは pktool(1) setpin コマンドを使用し、古いパスフレーズとしてデフォルトのパスフレーズ「changeme」を設定して、オブジェクトストアのパスフレーズを変更する必要があります。この操作は、パスフレーズを初期化して、新しく作成したトークンオブジェクトストアに設定するために必要です。

ユーザーは pktool setpin コマンドで設定した新しいパスフレーズを使用してオブジェクトストアにログインしたあと、この新しく作成されたオブジェクトストアに非公開のトークンオブジェクトを作成して保存できます。setpin でトークンオブジェクトストアを初期化するまでは、C_Login() 関数を使用することはできますが、ユーザーが非公開のトークンオブジェクトの作成、生成、派生、または検索を試みるとすべて失敗し、CKR_PIN_EXPIRED エラーが返されます。

C_Login() 関数と C_SetPIN() 関数に指定する PIN には、長さが 1 から 256 までで、空文字を含まない、任意の文字列を使用できます。

トークンオブジェクトストアのデフォルトの場所は /var/user/$USERNAME/pkcs11_softtoken です。

ユーザーは ${SOFTTOKEN_DIR} 環境変数を使用すると、デフォルトの場所を変更できます。代替のトークンオブジェクトストアの場所は ${SOFTTOKEN_DIR}/pkcs11_softtoken/ です。

戻り値

実装されている各関数の戻り値は、RSA PKCS#11 v2.20 仕様で定義されています。http://www.rsasecurity.com を参照してください。

ファイル

/var/user/$USERNAME/pkcs11_softtoken	ユーザーのデフォルトのトークンオブジェクトストア
${SOFTTOKEN_DIR}/pkcs11_softtoken	代替のトークンオブジェクトストア

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実
MT レベル	例外付きで MT-安全。RSA PKCS#11 v2.20 のセクション 6.6.2 を参照してください。
標準	PKCS#11 v2.20

関連項目 pktool(1), cryptoadm(1M), libpkcs11(3LIB), attributes(5), pkcs11_kernel(5)

RSA PKCS#11 v2.20 http://www.rsasecurity.com

名前

pkcs11_tpm – トラステッドプラットフォームモジュール (TPM) 用 RSA PKCS#11 トークン

形式

```
/usr/lib/security/pkcs11_tpm.so

/usr/lib/security/64/pkcs11_tpm.so
```

機能説明

pkcs11_tpm.so オブジェクトは、Trusted Computing Group のプロトコルを使用して TPM セキュリティーデバイスと通信することで、RSA Security Inc. の PKCS#11 暗号化トークンインタフェース (Cryptoki) v2.20 仕様を実装します。このプロバイダは PKCS#11 仕様を実装し、SUNWtss パッケージの TCG ソフトウェアスタック (TSS) API を使用します。

アプリケーション開発者は、pkcs11_tpm.so に直接リンクするのではなく、libpkcs11.so.1 にリンクするようにしてください。libpkcs11(3LIB) を参照してください。

実装されている暗号化アルゴリズムは、RSA、SHA1、および MD5 です。

libpkcs11(3LIB) のリストに示されている標準の PKCS#11 関数は、次を除きすべて実装されています。

```
C_EncryptUpdate
C_EncryptFinal
C_DecryptUpdate
C_DecryptFinal
C_DigestEncryptUpdate
C_DecryptDigestUpdate
C_SignEncryptUpdate
C_DecryptVerifyUpdate
C_GetFunctionStatus
C_CancelFunction
C_WaitForSlotEvent
C_GenerateKey
C_DeriveKey
```

サポートされている RSA PKCS#11 v2.20 メカニズムは次のとおりです。

```
CKM_RSA_PKCS_KEY_PAIR_GEN
CKM_RSA_PKCS
CKM_RSA_PKCS_OAEP
CKM_RSA_X_509
CKM_MD5_RSA_PKCS
CKM_SHA1_RSA_PKCS
CKM_SHA_1
CKM_SHA_1_HMAC
CKM_SHA_1_HMAC_GENERAL
CKM_MD5
CKM_MD5_HMAC
CKM_MD5_HMAC_GENERAL
```

ユーザーごとの初期化

pkcs11_tpm プロバイダを使用できるのは、システムに TPM デバイスが存在し、かつ SUNWtss パッケージがインストールされている場合だけです。これらの前提条件が満たされている場合、ユーザーは pktool(1) を使用すると自分専用のプライベートトークンを作成できます。このトークンにより、TPM デバイスを使用した操作を実行でき、TPM によって保護された鍵で自分の非公開データを保護できます。

ユーザーの TPM トークンを準備して初期化するには、次の手順を実行する必要があります。

1. トークンを初期化します。
2. SO (セキュリティー責任者) の PIN を設定します。
3. ユーザーの一意の PIN を設定します。

トークンを初期化するには、pktool(1) コマンドを次のように使用します。

```
$ pktool inittoken currlabel=TPM newlabel=tpm/myname
```

- デフォルトでは、初期化されていない TPM は TPM という名前で認識されます。ユーザーが自分専用のプライベートトークンを初期化するときに、別の名前 (tpm/joeuser など) に変更することも、TPM のままにする (その場合は newlabel 引数を省略する) こともできます。
- ユーザーは自分のトークンを初期化する前に、デフォルトの SO PIN を入力する必要があります。デフォルトの SO PIN は 87654321 です。これは上記の手順 2 で変更されています。

トークンを初期化したあとは、SO PIN とユーザー PIN をデフォルト値から変更する必要があります。これらの PIN の値を変更するには、再度 pktool(1) を使用します。

SO PIN の変更は次のように行います。

```
$ pktool setpin token=tpm/joeuser so
```

so オプションは、この「setpin」操作が SO PIN の変更であることを示すもので、省略できません。ユーザーはデフォルトの SO PIN (87654321) を入力してから、新しい PIN を入力 (および確認) する必要があります。

SO PIN をデフォルトから変更したあと、ユーザーの一意の PIN もリセットする必要があります。

ユーザーの PIN の変更は次のように行います。

```
$ pktool setpin token=tmp/joeuser
```

SO 以外のユーザーのデフォルト PIN は 12345678 です。ユーザーはデフォルトの PIN を入力してから、新しい一意の PIN を入力 (および確認) する必要があります。

pktool setpin 操作または C_Login() 関数と C_SetPIN() 関数に設定する PIN には、長さが 1 から 256 までで、空文字を含まない、任意の文字列を使用できます。

トークンへのアクセス

ユーザーは自分のトークンを初期化したあと、pktool(1) で、あるいは PKCS11 アプリケーションを作成することにより、トークンの使用を開始できます。その際、前述の手順で作成した名前 (前述の例では tpm/joeuser) を使用してトークンを特定します。

たとえば、

```
$ pktool gencert token=tpm/joeuser -i
$ pktool list token=tpm/joeuser
```

注意事項

pkcs11_tpm.so により、ファイルシステム固有のトークンオブジェクトの記憶領域にオブジェクトストレージが提供されます。プライベートオブジェクトは秘密鍵による暗号化で保護されています。復号化するには、トークンの秘密鍵を TPM に読み込み、復号化をすべて TPM 内で実行する必要があります。ユーザーの秘密鍵は、ユーザーが個人用の PIN を設定したときに (前述の説明を参照)、TPM によって生成されます。SO の鍵もユーザーの鍵も、TSS 永続的ストレージデータベースに保存され、一意の UUID 値で参照されます。どのユーザートークンにも一意の SO 鍵と一意のユーザー鍵があるため、あるユーザーのトークンの PIN で、同じマシンの別のユーザーのトークンに保存されている非公開データがロック解除されることはありません。

TPM はそれぞれ一意なので、ある TPM で作成されたトークンキーを別の TPM で使用することはできません。pkcs11_tpm.so トークンのデータは、TPM が存在しているシステムですべて管理されます。ほかのシステムに移動することはできません。TPM がリセットされ SRK (ストレージルートキー) が変更された場合、その TPM でそれまでに生成された鍵はすべて無効になります。

pkcs11_tpm.so は、作成されたトークンごとに専用のワークスペースを作成して、管理ファイルを管理します。デフォルトでは、この領域は /var/user/$USERNAME/tpm/ として作成されます。ただし、ユーザーはトークンの初期化や使用を開始する前に PKCS11_TPM_DIR 環境変数を設定することにより、この設定をオーバーライドできます。

戻り値

実装されている各関数の戻り値は、RSA PKCS#11 v2.20 仕様で定義されています。http://www.rsasecurity.com を参照してください。

ファイル

/var/user/$USERNAME/tpm/
　ユーザーのデフォルトのトークンオブジェクトストア。

${PKCS11_TPM_DIR}
　代替のトークンオブジェクトストア。

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実

属性タイプ	属性値
MT レベル	例外付きで MT-安全 (後述の説明を参照)
標準	PKCS#11 v2.20

「MT-安全」属性の例外については、RSA PKCS#11 v2.20 のセクション 6.6.2 を参照してください。

関連項目

`pktool(1)`, `cryptoadm(1M)`, `libpkcs11(3LIB)`, `attributes(5)`

TCG ソフトウェアスタック (TSS) 仕様 https://www.trustedcomputinggroup.org/specs/TSS (公開時点のアドレス)

RSA PKCS#11 v2.20 http://www.rsasecurity.com

名前

pkg – Image Packaging System

機能説明

Image Packaging System、pkg(5) は、ソフトウェアのライフサイクル管理 (インストール、アップグレード、および削除) のために提供されるフレームワークです。Image Packaging System は、一連のキーと値のペアおよび (場合によっては) データペイロードで定義されたアクションのコレクションである、パッケージの単位でソフトウェアを管理します。多くの場合、アクションはファイルシステム内に存在するファイルですが、ドライバ、サービス、ユーザーなどのその他のインストール可能オブジェクトも表します。

パッケージの **Fmri** とバージョン

各パッケージは、スキーム pkg: を使用して、障害管理リソース識別子 (FMRI) によって表されます。パッケージの完全な FMRI は、次の形式のスキーム、パブリッシャー、パッケージ名、およびバージョン文字列で構成されます。

```
pkg://solaris/system/library/c++-runtime@0.5.11,5.11-0.175.0.0.0.2.1:20120921T190358Z
```

solaris は発行元です。system/library/c++-runtime はパッケージ名です。名前空間は階層的であり、その深さは任意ですが、適用される包含関係は存在しません。名前は基本的に任意です。発行元情報はオプションですが、存在する場合は pkg:// を前に付ける必要があります。発行元を含む FMRI は多くの場合、「完全指定」と呼ばれます。発行元情報が存在しない場合は一般に、パッケージ名の前に pkg:/ が付けられます。

クライアントをパッケージ化すると、通常、FMRI にパブリッシャー情報が含まれていない場合に FMRI のスキームを省略できます。たとえば、pkg:/system/library/c++-runtime は system/library/c++-runtime と書くことができます。スキームが省略される場合、クライアントは照合の目的で、パッケージ名の最後のコンポーネントを除くすべてのコンポーネントを省略することもできます。たとえば、system/library/c++-runtime は library/c++-runtime または c++-runtime と書くことができ、これらは c++-runtime という名前のパッケージまたは /c++-runtime で終わるパッケージ名に一致します。

パブリッシャーの名前によって、人、人のグループ、または組織が 1 つ以上のパッケージのソースとして識別されます。パブリッシャーの名前の競合を避け、パブリッシャーを識別しやすくするために、パッケージを公開するエンティティーを表すドメイン名をパブリッシャーの名前として使用することがベストプラクティスです。

パッケージ名のあとに、アットマーク記号 (@) で区切られたバージョンが続きます。バージョンは、句読文字で区切られた 4 つの数字の並びで構成されます。最初の 3 つの並びに含まれる要素はドットで区切られ、これらの並びの長さは任意です。バージョンコンポーネント内の先頭のゼロ (たとえば、01.1 または 1.01) は許可されません。末尾のゼロ (たとえば、1.10) は許可されます。

バージョンの最初の部分はコンポーネントバージョンです。オペレーティングシステムに緊密に結合されたコンポーネントの場合、これは通常、そのバージョンのオ

ペレーティングシステムでの uname -r の値です。独自の開発ライフサイクルを持つコンポーネントの場合、この並びはドットで区切られたリリース番号 (2.4.10 など) です。

バージョンの 2 番目の部分 (存在する場合はコンマ (,) に続いている必要があります) はビルドバージョンです。ビルドバージョンは、パッケージの内容が構築されたオペレーティングシステムのバージョンを指定し、その内容が正常に実行されることを予測できるオペレーティングシステムのバージョンに関する最小の境界が提供されます。

バージョンの 3 番目の部分 (存在する場合はハイフン (-) に続いている必要があります) はブランチバージョンです。ブランチバージョンは、ベンダー固有の情報を提供するバージョン管理コンポーネントです。ブランチバージョンは、コンポーネントバージョンとは独立に、パッケージ化のメタデータが変更されたときに 1 増やすことができます。ブランチバージョンには、ビルド番号やその他の情報が含まれていることがあります。

バージョンの 4 番目の部分 (存在する場合はコロン (:) に続いている必要があります) はタイムスタンプです。タイムスタンプは、そのパッケージがいつ公開されたかを表します。

バージョン間の比較を実行する場合、その左側にあるコンポーネントが同じでないかぎり、バージョン全体のコンポーネントは考慮されません。したがって、「4.3」が「4.2」より大きいため「4.3-1」は「4.2-7」より大きく、「3」が「1」より大きいため「4.3-3」は「4.3-1」より大きくなります。

システムの多くの部分では、表示される情報量や必要な情報量を減らすために、必要に応じて FMRI を表示するときには短縮し、入力も短い形式を受け入れます。通常、スキーム、パブリッシャー、ビルドバージョン、およびタイムスタンプは省略できます。すべてのバージョン管理情報を省略できることもあります。

アクション

アクションはシステム上のインストール可能オブジェクトを表します。アクションはパッケージのマニフェストに記述されます。すべてのアクションは、最初に名前とキー属性を含みます。また、これらはバージョン履歴をたどる過程で一意のオブジェクトを参照します。アクションには、このほかの属性も含まれます。一部の属性は、パッケージシステムによって直接解釈されます。その他の属性は、システム管理者またはエンドユーザーにのみ役立つ可能性があります。

アクションには、次の単純なテキスト表現があります。

action_name attribute1=value1 attribute2=value2 ...

属性の名前に、空白、引用符、または等号 (=) を含めることはできません。最初の等号のあとの文字はすべて、値に属します。値にはすべての文字を含めることができますが、スペースは単一引用符または二重引用符で囲む必要があります。二重引用符で囲まれている文字列の内部で単一引用符をエスケープする必要はなく、単一引用符で囲まれている文字列の内部で二重引用符をエスケープする必要はありませ

ん。引用符の前にバックスラッシュ (\) 文字を付けて、引用文字列が終了しないようにすることができます。バックスラッシュは、バックスラッシュでエスケープできます。

複数の値を使用して、属性に複数回名前を付けることができます。これらは、順序なしリストとして扱われます。

多くの属性を持つアクションによって、マニフェストファイル内に長い行が作成されることがあります。このような行は、不完全な各行をバックスラッシュで終了することにより折り返すことができます。この継続文字は、属性と値のペアの間に置く必要があることに注意してください。属性も、その値も、さらにその組み合わせも分割することはできません。

下に示されている属性セットは、これがすべてではありません。実際、アクションに付加できる属性は任意であるため、標準の属性セットは、将来の開発を実装するために容易に拡張できます。

特定のアクション属性によって、パッケージ化コンテキストの外部で追加の操作が実行されます。これらの属性は、下の「アクチュエータ」のセクションで説明されています。

ファイルアクション

`file` アクションは、通常ファイルを表します。`file` アクションはペイロードを参照し、次の 4 つの標準属性があります。

`path` ファイルがインストールされているファイルシステムのパス。これは `file` アクションのキー属性です。

`mode` ファイルのアクセス権 (数値形式)。これらは ACL ではなく、単純なアクセス権のみです。

`owner` ファイルを所有するユーザーの名前。

`group` ファイルを所有するグループの名前。

ペイロードは、名前が付けられないという点で位置属性です。これは、アクション名のあとの最初の単語です。公開されたマニフェストでは、これはファイルの内容の `SHA-1` ハッシュです。これから公開する必要のあるマニフェスト内に存在する場合は、ペイロードを見つけることのできるパスを表します。`pkgsend(1)` を参照してください。値に等号が含まれている場合は、位置属性の代わりにハッシュ属性を使用できます。この両方を同じアクションで使用できます。ただし、ハッシュは同じである必要があります。

その他の属性には次のものがあります。

`preserve`
これは、ファイルの内容がそのファイルのインストールまたは最後のアップグレード以降に変更されたと判定された場合、アップグレード時にその内容を上書

きすべきではないことを指定します。初期インストール時に既存のファイルが見つかった場合、そのファイルは回収されます (/var/pkg/lost+found 内に格納されます)。

preserve の値が renameold である場合は、既存のファイルの名前が拡張子 .old を使用して変更され、新しいファイルがその場所に置かれます。

preserve の値が renamenew である場合は、既存のファイルがそのままになり、新しいファイルが拡張子 .new を使用してインストールされます。

preserve の値が legacy である場合は、初期のパッケージインストールでこのファイルはインストールされません。アップグレード時、既存のファイルはすべて拡張子 .legacy を使用して名前が変更され、新しいファイルがその場所に置かれます。

preserve の値が true (または、strawberry などの上に示されていない値) である場合は、既存のファイルがそのままになり、新しいファイルはインストールされません。

overlay
これは、このアクションによってほかのパッケージが同じ場所にファイルを提供できるか、またはこのアクションによって別のファイルをオーバーレイすることを目的にしたファイルが提供されるかを指定します。この機能は、どの自己アセンブリにも参加しておらず (たとえば、/etc/motd)、かつ安全に上書きできる構成ファイルで使用されることを目的にしています。

overlay が指定されていない場合は、複数のパッケージが同じ場所にファイルを提供することはできません。

overlay の値が allow である場合は、ほかの 1 つパッケージが同じ場所にファイルを提供することを許可されます。preserve 属性も同時に設定されていないかぎり、この値は意味を持ちません。

overlay の値が true である場合は、このアクションによって提供されたファイルにより、allow を指定したほかのアクションがすべて上書きされます。インストールされたファイルへの変更は、オーバーレイしているファイルの preserve 属性の値に基づいて保持されます。削除時、ファイルの内容は、preserve 属性が指定されたかどうかには関係なく、オーバーレイされているアクションがまだインストールされている場合は保持されます。別のファイルをオーバーレイできるのは 1 つのアクションだけであり、mode、owner、および group 属性が一致している必要があります。

ファイルは「味見」してみることもできます。その「風味」に応じて、追加で属性を付けることができます。ELF ファイルの場合は、次の属性が認識されます。

elfarch
ELF ファイルのアーキテクチャー。これは、そのファイルが構築されたアーキテクチャー上での uname -p の出力です。

elfbits
これは 32 または 64 です。

elfhash
これは、そのファイル内の「興味深い」ELF セクションのハッシュです。これらは、バイナリがロードされるときにメモリーにマップされるセクションです。これらは、2 つのバイナリの実行可能ファイルの動作が異なるかどうかを判定するときに考慮する必要のある唯一のセクションです。

original_name
この属性は、パッケージからパッケージに、または場所から場所に、あるいはその両方で移動している編集可能なファイルを処理するために使用されます。この形式は、元のパッケージの名前のあとに、コロンとこのファイルの元のパスが続きます。削除されているファイルはすべて、そのパッケージとパス、または original_name 属性の値 (指定されている場合) のどちらかを使用して記録されます。original_name 属性が設定された、インストールされている編集可能なファイルはすべて、それが同じパッケージ化の操作の一部として削除されている場合は、その名前のファイルを使用します。

release-note
この属性は、このファイルにリリースノートテキストが含まれていることを示すために使用されます。この属性の値は、パッケージ FMRI です。元のイメージに存在し、元のイメージ内のパッケージよりも新しいバージョンのパッケージ名が FMRI で指定されている場合、このファイルはリリースノートに含まれます。特別な FMRI (feature/pkg/self) は、含まれるパッケージを指します。feature/pkg/self のバージョンが 0 の場合、このファイルは初期インストールのリリースノートにのみ含まれます。

revert-tag
この属性は、セットとして元に戻すべき編集可能なファイルをタグ付けするために使用されます。複数の revert-tag 値を指定できます。これらのいずれかのタグを指定して pkg revert が呼び出されると、ファイルはマニフェストで定義された状態に戻ります。pkg(1) を参照してください。

ディレクトリアクション

dir アクションは、ファイルシステムオブジェクトを表すという点で file アクションに似ています。dir アクションは、通常ファイルの代わりにディレクトリを表します。dir アクションには、file アクションと同じ 4 つの標準属性があり、path がキー属性です。

ディレクトリは、IPS でカウントされる参照です。あるディレクトリを明示的または暗黙的に参照している最後のパッケージが参照を行わなくなると、そのディレクトリは削除されます。そのディレクトリにパッケージ解除されたファイルシステムオブジェクトが含まれている場合、それらの項目は $IMAGE_META/lost+found に移動されます。$IMAGE_META についての詳細は、「ファイル」のセクションを参照してください。

パッケージ解除された内容を新しいディレクトリに移動するために、次の属性が役立つことがあります。

`salvage-from`
これは、回収された項目のディレクトリを指定します。このような属性を持つディレクトリは、作成時、回収されたディレクトリの内容を継承します (その内容が存在する場合)。

リンクアクション

`link` アクションはシンボリックリンクを表します。`link` アクションには、次の標準属性があります。

`path`
シンボリックリンクがインストールされるファイルシステムのパス。これは `link` アクションのキー属性です。

`target`
シンボリックリンクのターゲット。リンクの解決先のファイルシステムオブジェクト。

`mediator`
特定のメディエーショングループ (たとえば、`python`) に参加しているすべてのパス名によって共有されているメディエーション名前空間内のエントリを指定します。`mediator-version` または `mediator-implementation`、あるいはその両方に基づいてリンクメディエーションを実行できます。特定のパス名に対してメディエートされたリンクはすべて、同じメディエータを指定する必要があります。ただし、すべてのメディエータバージョンおよび実装が、特定のパスでリンクを提供する必要はありません。メディエーションがリンクを提供していない場合は、そのメディエーションが選択されたときにリンクが削除されます。特定のバージョンまたは実装、あるいはその両方と組み合わせた `mediator` は、パッケージシステムで使用するために選択できるメディエーションを表します。

`mediator-version`
`mediator` 属性で記述されたインタフェースの (負にならない整数のドットで区切られた並びとして表された) バージョンを指定します。この属性は、`mediator` が指定され、`mediator-implementation` は指定されていない場合に必要です。ローカルシステム管理者は、使用するバージョンを明示的に設定できます。指定された値は一般に、リンクを提供しているパッケージのバージョンに一致するようにしてください (たとえば、`runtime/python-26` は `mediator-version=2.6` を使用します)。ただし、これは必須ではありません。

`mediator-implementation`
`mediator-version` に加えて、またはこの代わりに使用するメディエータの実装を指定します。実装の文字列は順序付けられているとは見なされず、システム管理者によって明示的に指定されていない場合は、`pkg` (5) によって文字列が任意に選択されます。

この値は、英数字とスペースで構成された任意の長さの文字列にすることができます。実装自体をバージョン管理できるか、または実際にバージョン管理されている場合は、文字列の最後の @ のあとに (負にならない整数のドットで区切られた並びとして表された) バージョンを指定します。実装の複数のバージョンが存在する場合、デフォルトの動作では、最大のバージョンを持つ実装が選択されます。

特定のパスにある実装メディエーションリンクの 1 つのインスタンスだけがシステムにインストールされている場合は、その 1 つのインスタンスが自動的に選択されます。このパスにある将来のリンクがインストールされても、ベンダー、サイト、またはローカルのオーバーライドが適用されないかぎり、またはいずれかのリンクのバージョンがメディエートされた場合、このリンクは切り替えられません。

`mediator-priority`
メディエートされたリンクの競合を解決する場合、pkg(5) は通常、`mediator-version` の最大の値を持つリンクを選択するか、またはそれが不可能な場合は `mediator-implementation` に基づいて選択します。この属性は、正常な競合解決処理のためのオーバーライドを指定するために使用されます。

この属性が指定されていない場合は、デフォルトのメディエータ選択ロジックが適用されます。

この値が `vendor` である場合は、このリンクが、`mediator-priority` が指定されていないリンクより優先されます。

この値が `site` である場合は、このリンクが、`vendor` の値を持つリンクや、`mediator-priority` が指定されていないリンクより優先されます。

ローカルシステム管理者は、上で説明した選択ロジックをオーバーライドできます。

ハードリンクアクション

`hardlink` アクションは、ハードリンクを表します。このアクションは `link` アクションと同じ属性を持ち、そのキー属性も同じく `path` です。

ドライバアクション

`driver` アクションはデバイスドライバを表します。`driver` アクションはペイロードを参照しません。ドライバファイル自体を `file` アクションとしてインストールする必要があります。次の属性が認識されます (詳細は `add_drv`(1M) を参照)。

`name`
ドライバの名前。多くの場合はドライババイナリのファイル名ですが、必ずしもそうとは限りません。これは `driver` アクションのキー属性です。

`alias`
これはドライバの別名を表します。特定のドライバが複数の `alias` 属性を持つことができます。特殊な引用符の規則は必要ありません。

class
これはドライバクラスを表します。特定のドライバが複数の class 属性を持つことができます。

perms
これは、ドライバのデバイスノードのファイルシステムアクセス権を表します。

clone_perms
これは、このドライバに対する複製ドライバのマイナーノードのファイルシステムアクセス権を表します。

policy
これは、デバイスのための追加のセキュリティーポリシーを指定します。特定のドライバが複数の policy 属性を持つことができますが、マイナーデバイスの指定が複数の属性に存在することはできません。

privs
これは、ドライバで使用される特権を指定します。特定のドライバが複数の privs 属性を持つことができます。

devlink
これは /etc/devlink.tab 内のエントリを指定します。この値はファイルに書き込まれる行そのものであり、タブは \t で示されます。詳細は、devlinks(1M) を参照してください。特定のドライバが複数の devlink 属性を持つことができます。

依存アクション

depend アクションは、パッケージ間の依存関係を表します。あるパッケージが別のパッケージに依存することがあります。たとえば、最初のパッケージの機能を有効にする (場合によってはインストールする) ために、2 番目のパッケージの機能が必要な場合があります。依存関係は省略可能です。インストール時に依存関係が満たされない場合、パッケージシステムはほかの制約に応じて、依存パッケージをインストールするか、または十分に新しいバージョンに更新しようとします。

次の属性が認識されます。

fmri
依存パッケージを表す FMRI。これは dependency アクションのキー属性です。fmri 値にパブリッシャーを含めることはできません。パッケージ名は完全であると見なされます。タイプ require-any の依存関係は、複数の fmri 属性を持つことができます。fmri 値でバージョンは省略可能ですが、依存関係のタイプによっては、バージョンのない fmri は無意味である場合があります。

type
依存関係のタイプ。

require
依存関係が必要であり、かつ fmri 属性で指定されたバージョン以上のバージョンを持っている必要があります。バージョンが指定されていない場

合は、任意のバージョンが依存関係を満たします。パッケージの必要な依存関係のいずれかを満たすことができない場合、そのパッケージはインストールできません。

optional
依存関係が存在する場合は、指定されたバージョンレベル以上である必要があります。

exclude
指定されたバージョンレベル以上という依存関係が存在する場合は、含んでいるパッケージをインストールできません。バージョンが指定されていない場合、依存パッケージは、依存関係を指定しているパッケージと同時にインストールできません。

incorporate
依存関係はオプションですが、依存するパッケージのバージョンは制約されます。以下の「制約と凍結」を参照してください。

require-any
複数の fmri 属性で指定された複数の依存パッケージのうちのいずれか 1 つが、依存関係のタイプ require と同じ規則に従って依存関係を満たすことができます。

conditional
依存関係は、predicate 属性で定義されたパッケージがシステム上に存在する場合にのみ必要です。

origin
依存関係が存在する場合、依存関係はインストールの前に変更されるイメージ上の指定された値以上である必要があります。root-image 属性の値が true である場合、このパッケージをインストールするには、/ をルートとするイメージ上に依存関係が存在する必要があります。

group
依存関係は、パッケージがイメージ回避リスト上にないかぎり必要です。廃止されたパッケージは、暗黙のうちにグループの依存関係を満たすことに注意してください。pkg(1) の avoid サブコマンドを参照してください。

parent
依存関係は、イメージが子イメージでない場合は無視されます。このイメージが子イメージである場合は、親イメージ内に依存関係が存在する必要があります。parent 依存関係でのパッケージバージョンの照合は、incorporate 依存関係で使用されるものと同じです。

predicate
conditional 依存関係の述語を表す FMRI。

root-image
先に説明した origin 依存関係に対してのみ有効です。

ライセンスアクション

license アクションは、パッケージの内容に関連したライセンスやその他の情報ファイルを表します。パッケージは license アクションの使用を通して、ライセンス、免責条項、またはその他のガイダンスをパッケージインストーラに提供できます。

license アクションのペイロードは、パッケージに関連したイメージメタデータディレクトリに提供され、人間が読める形式のテキストデータのみが含まれているべきです。HTML やその他の形式のマークアップが含まれていてはいけません。license アクションは、属性を通して、関連するペイロードが表示または同意、あるいはその両方を必要としていることをクライアントに示すことができます。表示または同意、あるいはその両方の方法は、クライアントに任されています。

次の属性が認識されます。

license
これは license アクションのキー属性です。この属性は、ユーザーがライセンスのテキスト自体を読まなくてもその内容を判断できるように支援するための、ライセンスの意味のある説明を提供します。この値のいくつかの例を次に示します。

- ABC Co. Copyright Notice
- ABC Co. Custom License
- Common Development and Distribution License 1.0 (CDDL)
- GNU General Public License 2.0 (GPL)
- GNU General Public License 2.0 (GPL) Only
- MIT License
- Mozilla Public License 1.1 (MPL)
- Simplified BSD License

license 値は、パッケージ内で一意である必要があります。上のいくつかの例に示すように、説明にライセンスのバージョンを含めることをお勧めします。パッケージに複数のライセンスに基づくコードが含まれる場合、複数の license アクションを使用します。ライセンス属性値の長さは、64 文字以下にしてください。

must-accept
true の場合は、関連するパッケージをインストールまたは更新するには、ユーザーがこのライセンスに同意する必要があります。この属性を省略すると、false と同等になります。同意の方法 (たとえば、対話型または構成ベース) は、クライアントに任されています。

must-display
true の場合は、パッケージ化の操作中に、クライアントがこのアクションのペイロードを表示する必要があります。この値を省略すると、false と同等になります。この属性は、コピーライト表示に使用してはいけません。操作中に表示する必要のある実際のライセンスまたはその他の素材にのみ使用してください。表

示の方法は、クライアントに任されています。

レガシーアクション

legacy アクションは、従来のパッケージシステムで使用されるパッケージデータを表します。このアクションに関連付けられた属性は、従来のシステムのデータベースに追加されます。そのため、これらのデータベースに問い合わせを行なっているツールは、従来のパッケージが実際にインストールされているかのように動作できます。特にこれにより、pkg 属性で指定されたパッケージがシステムにインストールされていることを従来のシステムに十分に確信させることができるため、このパッケージを、依存関係を満たすために使用できるようになります。

pkginfo(4) のパラメータに従って指定された次の属性が認識されます。

category
CATEGORY パラメータの値。デフォルト値は system です。

desc
DESC パラメータの値。

hotline
HOTLINE パラメータの値。

name
NAME パラメータの値。デフォルト値は none provided です。

pkg
インストールされるパッケージの略語。デフォルト値は、パッケージの FMRI の名前です。これは legacy アクションのキー属性です。

vendor
VENDOR パラメータの値。

version
VERSION パラメータの値。デフォルト値は、パッケージの FMRI のバージョンです。

設定アクション

set アクションは、パッケージの説明などの、パッケージレベルの属性 (またはメタデータ) を表します。

次の属性が認識されます。

name　属性の名前。

value　属性に与えられた値。

set アクションは、パッケージ作成者が選択した任意のメタデータを提供できます。ただし、パッケージシステムにとって特別な意味を持つ、適切に定義された属性名がいくつか存在します。

pkg.fmri
「説明」セクションの「パッケージの FMRI とバージョン」を参照してください。

`info.classification`
: pkg(5) クライアントがパッケージを分類するために使用できる 1 つ以上のトークン。この値には、スキーム (「org.opensolaris.category.2008」や「org.acm.class.1998」など) と実際の分類 (「アプリケーション/ゲーム」など) がコロン (:) で区切られて含まれています。

`pkg.description`
: パッケージの内容と機能の詳細な説明。通常は、1 つの段落程度の長さです。

`pkg.obsolete`
: `true` の場合、パッケージは廃止としてマークされています。廃止されたパッケージには、ほかの設定アクション以外のアクションは存在せず、また名前変更としてマークすることはできません。

`pkg.renamed`
: `true` の場合は、パッケージの名前が変更されました。このパッケージには、このパッケージの名前が変更された先のパッケージバージョンを指す 1 つ以上の `depend` アクションも存在する必要があります。パッケージを名前変更と廃止の両方としてマークすることはできませんが、それ以外は任意の数の設定アクションが存在できます。

`pkg.summary`
: パッケージの短い、1 行の説明。

グループアクション

`group` アクションは、`group`(4) で定義されるのと同様の UNIX グループを定義します。グループパスワードのサポートはありません。このアクションで定義されたグループには最初、ユーザーリストがありません。ユーザーは、`user` アクションを使用して追加できます。次の属性が認識されます。

`groupname`
: グループの名前の値。

`gid`
: グループの一意の数値 ID。デフォルト値は、100 未満の最初に空いているグループです。

ユーザーアクション

`user` アクションは、`/etc/passwd`、`/etc/shadow`、`/etc/group`、および `/etc/ftpd/ftpusers` ファイルで定義されるのと同様の UNIX ユーザーを定義します。このアクションでユーザーを定義すると、しかるべきファイルにエントリが追加されます。

次の属性が認識されます。

`username`
: ユーザーの一意の名前。

`password`
: ユーザーの暗号化パスワード。デフォルト値は `*LK*` です。`shadow`(4) を参照してください。

uid
ユーザーの一意の UID。デフォルト値は、100 未満の最初に空いている値です。

group
ユーザーのプライマリグループの名前。/etc/group に存在する必要があります。

gcos-field
/etc/passwd 内の gcos フィールドの値。デフォルト値は username です。

home-dir
ユーザーのホームディレクトリ。デフォルト値は / です。

login-shell
ユーザーのデフォルトのシェル。デフォルト値は空です。

group-list
ユーザーが属しているセカンダリグループ。group(4) を参照してください。

ftpuser
true または false に設定できます。true のデフォルト値は、ユーザーが FTP 経由のログインを許可されていることを示します。ftpusers(4) を参照してください。

lastchg
1970 年 1 月 1 日と、パスワードが最後に変更された日付の間の日数。デフォルト値は空です。shadow(4) を参照してください。

min
パスワード変更の間の必要な最小日数。パスワードの有効期限を有効にするには、このフィールドを 0 以上に設定する必要があります。デフォルト値は空です。shadow(4) を参照してください。

max
パスワードが有効な最大日数。デフォルト値は空です。shadow(4) を参照してください。

warn
パスワードの期限が切れる前にユーザーに警告が表示される日数。shadow(4) を参照してください。

inactive
そのユーザーに許可されている非活動の日数。これはマシンごとにカウントされます。最終ログインに関する情報は、そのマシンの lastlog ファイルから取得されます。shadow(4) を参照してください。

expire
UNIX エポック (1970 年 1 月 1 日) 以降の日数として表される絶対的な日付。この日数に達すると、ログインを使用できなくなります。たとえば、13514 の期限切れの値は、ログインの有効期限が 2007 年 1 月 1 日であることを指定します。shadow(4) を参照してください。

`flag`
空に設定されています。`shadow`(4) を参照してください。

アクチュエータ

コンテキストによっては、特定のアクションの準備として、またはその導入のあとに、追加の操作を実行することが適している場合があります。これらの追加の操作は一般に、ライブシステムイメージ上でのみ必要であり、またオペレーティングシステムに固有のものです。パッケージのインストールまたは削除に関与する複数のアクションのアクチュエータが同一である場合、アクチュエータの存在に対応する操作は、そのインストールまたは削除に対して 1 回実行されます。

アクチュエータを誤って指定すると、そのアクチュエータがインストールを安全に進めるための手段を決定できない場合、パッケージのインストールが失敗することがあります。

次のアクチュエータが定義されています。

`reboot-needed`
`true` または `false` に設定できます。このアクチュエータは、パッケージシステムがライブイメージ上で動作している場合、タグ付きアクションの更新または削除を新しいブート環境で実行する必要があることを宣言します。新しいブート環境の作成は、`be-policy` イメージプロパティーによって制御されます。`be-policy` プロパティーについての詳細は、`pkg`(1) のマニュアルページの「イメージプロパティー」セクションを参照してください。

`disable_fmri`、`refresh_fmri`、`restart_fmri`、`suspend_fmri`
これらの各アクチュエータは、パッケージのインストールまたは削除中に操作するサービスインスタンスの FMRI の値を取ります。`disable_fmri` を指定すると、`svcadm`(1M) の `disable` サブコマンドに従って、アクションの削除の前に特定の FMRI が無効になります。`refresh_fmri` および `restart_fmri` を指定すると、`svcadm`(1M) の対応するサブコマンドに従って、アクションのインストール、更新、または削除のあとに特定の FMRI が更新または再起動されます。最後に、`suspend_fmri` を指定すると、アクションのインストールフェーズの前に特定の FMRI が一時的に無効になり、そのフェーズが完了したあとに有効になります。

この値には、複数のサービスインスタンスに一致するパターンを含めることができます。ただし、それをインスタンスを示さずに暗黙的に行うのではなく、`svcs`(1) によって受け入れられた glob を使用して明示的に行う必要があります。

メディエーション

メディエータは、一連の関連するシンボリックリンクまたはハードリンクを表す名前です。2 つ以上のリンクアクションが同じパスとメディエータ名を持つ場合、ユーザーまたはパッケージシステムは、バージョン、実装、または優先順位に基づいてリンクターゲットを選択します。メディエータ属性については、「リンクアクション」を参照してください。

次の例は、リンクがバージョンによって選択される、java という名前のメディエータの 2 つの異なるインスタンスを示しています。これらの 2 つの link アクションは 2 つの異なるパッケージに出現します。

```
link mediator=java mediator-version=1.6 path=usr/java target=jdk/jdk1.6.0_31
link mediator=java mediator-version=1.7 path=usr/java target=jdk/jdk1.7.0_02
```

このリンクパスのバージョンを選択する方法については、pkg(1) のマニュアルページの set-mediator サブコマンドを参照してください。バージョンを選択するには、両方のパッケージがインストールされている必要があります。

制約と凍結

パッケージが新しいバージョンに移行される場合、またはシステムに追加されたり、システムから削除されたりする場合、選択されるバージョンや、削除が許可されるかどうかは、そのパッケージに対するさまざまな制約によって決定されます。これらの制約は、依存関係の形式でほかのパッケージが定義するか、または凍結の形式で管理者が定義できます。

制約のもっとも一般的な形式は、上の「依存アクション」で説明したように、require 依存関係によって提供されます。このような制約によって、パッケージがダウングレードまたは削除されることが回避されます。

オペレーティングシステムのほとんどの部分は、*incorporation* と呼ばれるパッケージによってカプセル化されています。これらのパッケージは主に、incorporate 依存関係によって表される制約を提供します。

上で説明したように、組み込まれたパッケージがシステム上に存在する必要はありませんが、存在する場合は、包括的な最小バージョンと排他的な最大バージョンの両方を指定します。たとえば、依存する FMRI のバージョンが 1.4.3 の場合、1.4.3 未満のバージョンは依存関係を満たさず、1.4.4 以上のバージョンでも依存関係を満たしません。ただし、1.4.3.7 など、単にドット形式の数列を拡張したバージョンは許可されます。

incorporation は、システムの各部の同期的なアップグレードを強制的に行うために使用されます。C ライブラリやカーネルなどの一部のコンポーネントでは、これは基本的な要件です。ほかには依存関係が存在しない単純なユーザーランドコンポーネントなどのその他のコンポーネントでは、incorporation の特定のバージョンから参照できるテストされた、既知の一連のパッケージバージョンを提供するためだけに、同期アップグレードが使用されます。

incorporation は単なるパッケージであるため、削除することができ、それが提供しているすべての制約がそれによって緩和されます。ただし、その緩和は安全ではないため、Oracle Solaris によって提供される多くの incorporation が、それらの incorporation によって組み込まれているパッケージには必要です。

パッケージを、インストール済みの incorporation によって許可されていないバージョンにアップグレードしようとしても、要求を満たすためにその incorporation の新しいバージョンを見つけようとする試みは行われず、そのアップ

グレードは失敗します。制約自体を移動する必要があるが、それを指定している incorporation を削除できない場合は、その incorporation を、制約の目的のバージョンを指定するバージョンにアップグレードする必要があります。incorporation をアップグレードすると、その新しいバージョンによって提供される制約を満たさない組み込まれたパッケージもすべてアップグレードされます。

システム管理者は、`pkg freeze` コマンドを使用してパッケージを制約できます。バージョンが指定されていない場合、指定されたパッケージは、システムにインストールされているバージョンに制約されます。バージョン管理されたパッケージが指定された場合、この管理上の制約 (つまり、凍結) は、`fmri` 属性が指定されたパッケージバージョンの値を持った状態で incorporate 依存関係がインストールされているかのように機能します。

凍結がパッケージシステムによって自動的に解除されることはありません。制約を緩和するには、`pkg unfreeze` コマンドを使用します。

発行元とリポジトリ

上で詳細に説明したように、パブリッシャーとは単に、パッケージクライアントがパッケージのプロバイダを識別するために使用する名前です。パブリッシャーは、パッケージリポジトリまたはパッケージアーカイブ、あるいはその両方を使用してパッケージを配布できます。現在パッケージシステムでサポートされているリポジトリのタイプには、起点リポジトリとミラーリポジトリの 2 つがあります。

起点は、1 つ以上のパッケージのすべてのメタデータ (カタログ、マニフェスト、検索インデックスなど) と内容 (ファイル) を含むパッケージリポジトリです。イメージ内の特定のパブリッシャーに対して複数の起点が構成されている場合、パッケージクライアント API は、パッケージデータの取得元として最適な起点を選択しようとします。これはもっとも一般的なタイプのリポジトリであり、パッケージリポジトリ上で `pkgsend` または `pkgrecv` が使用された場合は常に、暗黙的に作成されます。

ミラーは、パッケージの内容 (ファイル) のみを含むパッケージリポジトリです。イメージ内の特定のパブリッシャーに対して 1 つ以上のミラーが構成されている場合、クライアント API はパッケージ内容の取得のためのミラーを優先し、パッケージの内容の取得元として最適なミラーを選択しようとします。ミラーが到達不可能か、必要な内容が含まれていないか、またはより低速な場合、クライアント API は、構成されているいずれかの起点リポジトリから内容を取得します。ミラーは、`pkg.depotd`(1M) の動的ミラー機能を使用している、信頼できる一連のクライアント間の内容共有に使用されることを目的にしています。ミラーはまた、パッケージのメタデータへのアクセスを認証するために使用されることも目的にしています。ただし、パッケージの内容は認証なしで配布します。たとえば、あるクライアントが、アクセスするには SSL キーと証明書のペアが必要な `https` 起点と、パッケージの内容を提供する `http` ミラーを使用して構成される可能性があります。このようにして、認可クライアントだけがパッケージをインストールまたは更新できるようにしながら、パッケージ内容の取得のための認証のオーバーヘッドが回避されます。ミラーは、`file` という名前のサブディレクトリとその親を除

く、リポジトリのすべてのサブディレクトリを削除することによって作成できます。また、pkg.depotd(1M) のミラーモードを使用すると、起点リポジトリもミラーとしてプロビジョニングできます。

大域ゾーンと非大域ゾーンの更新

pkg システムは、非大域ゾーンを強制的に大域ゾーンと同期された状態に保持します。つまり、同じカーネルが確実に実行されるようにするために、特定のパッケージを大域ゾーンとすべての非大域ゾーンで同じバージョンにする必要があります。このために、pkg は parent 依存関係を使用して、非大域ゾーンに特定の制約を課します。parent 依存関係についての詳細は、上の「依存アクション」を参照してください。

大域ゾーンが非大域ゾーンに課す制限のため、非大域ゾーンは大域ゾーンのパッケージにアクセスできる必要があり、かつ同様の発行元構成にする必要があります。これらの目的はどちらも、システムリポジトリ を使用して実現されます (pkg.sysrepo(1M) のマニュアルページを参照)。システムリポジトリは、大域ゾーン内で構成された発行元へのアクセスと、これらの発行元が構成されている方法に関する情報を提供します。インストールまたは更新中に非大域ゾーンで別のパッケージが選択されないようにするために、発行元の検索順序で、システム発行元には非大域ゾーン内で構成された発行元より高くランク付けされます。発行元の検索順序については、pkg(1) のマニュアルページの pkg set-publisher コマンドを参照してください。

システム上のすべての非大域ゾーンを更新するには、大域ゾーン内で pkg update コマンドを引数なしで使用します。このコマンドは大域ゾーンに対して、また各非大域ゾーンに対して再帰的に動作します。非大域ゾーンを大域ゾーン内で行われた変更と同期された状態にするために、非大域ゾーンに対して必要な最小の変更が加えられます。たとえば、パッケージ foo が大域ゾーンと非大域ゾーンの両方にバージョン 1 でインストールされており、システムリポジトリ内でバージョン 2 が使用可能であると想定します。foo に親の依存関係が存在する場合、parent 依存関係によってパッケージは強制的に同期された状態に保持されるため、pkg update foo は foo を大域ゾーンと非大域ゾーンの両方でバージョン 2 に更新します。foo に親の依存関係が存在しない場合、foo は大域ゾーンでバージョン 2 に更新されますが、非大域ゾーンではバージョン 1 のままになります。

ファセットとバリアント

ソフトウェアには、省略可能なコンポーネントや、相互に排他的なコンポーネントが含まれることがあります。省略可能なコンポーネントの例には、ロケールやドキュメントがあります。相互に排他的なコンポーネントの例には、SPARC バイナリと x86 バイナリや、デバッグバイナリと非デバッグバイナリなどがあります。

IPS では、省略可能なコンポーネントをファセット、相互に排他的なコンポーネントをバリアントと呼びます。ファセットとバリアントはパッケージアクションのタグとして指定します。各ファセットタグおよびバリアントタグには名前と値があります。1 つのアクションに複数のファセットタグおよびバリアントタグを付けることができます。複数のファセットおよびバリアントタグのあるコンポーネントの例には、開発者によって使われるアーキテクチャー固有のヘッダーファイルや SPARC 大域ゾーン専用のコンポーネントなどがあります。

バリアントタグの例は variant.arch=sparc です。ファセットタグの例は facet.devel=true です。ファセットとバリアントは、facet. や variant. を先頭に付けずに参照されることがよくあります。

ファセットとバリアントはイメージの特殊なプロパティーであり、個々のパッケージには設定できません。イメージに設定されたファセットおよびバリアントの現在の値を表示するには、pkg(1) のマニュアルページに示すように、pkg facet コマンドと pkg variant コマンドを使用します。イメージに設定されたファセットおよびバリアントの値を変更するには、pkg change-facet コマンドと pkg change-variant コマンドを使用します。

ファセットはブール型です。それらには true (有効) または false (無効) のみ設定できます。デフォルトで、イメージ内のすべてのファセットは、true に設定されていると見なされます。アクションのファセットタグの値には true のみを指定すべきであり、それ以外の値では動作が不確定になります。イメージに設定されるファセットは、doc.man などの完全なファセットか、locale.* などのパターンになります。これは、ファセット名前空間の一部を無効にし、その中の個々のファセットのみを有効にする場合に役立ちます。たとえば、次の例に示すように、すべてのロケールを無効にしてから、1 つか 2 つの特定のロケールのみを有効にすることができます。

```
# pkg change-facet locale.*=false
[output about packages being updated]
# pkg change-facet locale.en_US=true
[output about packages being updated]
```

ほとんどのバリアントは任意の数の値を設定できます。たとえば、arch バリアントには、i386、sparc、ppc、arm、またはディストリビューションがサポートしているどのようなアーキテクチャーでも設定できます。(Oracle Solaris では i386 と sparc のみが使用されます。)例外は debug バリアントです。debug バリアントは、true または false のみ設定でき、ほかの値では動作が不定になります。ファイルアクションに非デバッグバージョンとデバッグバージョンの両方がある場合、次の例に示すように、両方のバージョンに該当する debug バリアントが明示的に設定されている必要があります。

```
file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=115 pkg.size=103 preserve=true \
  variant.debug.osnet=true

file group=sys mode=0644 overlay=allow owner=root \
  path=etc/motd pkg.csize=68 pkg.size=48 preserve=true \
  variant.debug.osnet=false
```

バリアントを使用するパッケージをインストールするために、バリアント値をイメージに設定する必要があります。arch および zone バリアントは、イメージを作成し、その初期コンテンツをインストールするプログラムによって設定されます。イメージ内の debug.* バリアントはデフォルトで false です。

イメージに設定されたファセットとバリアントは、特定のアクションがインストールされるかどうかに影響します。

- ファセットまたはバリアントタグのないアクションは常にインストールされます。
- ファセットタグのあるアクションは、イメージ上のタグに一致するすべてのファセットまたはファセットパターンが false に設定されていないかぎり、インストールされます。いずれかのファセットが true に設定されているか、明示的に設定されていない (true はデフォルト) 場合、アクションがインストールされます。
- バリアントタグのあるアクションは、すべてのバリアントタグの値がイメージに設定されている値と同じ場合にのみインストールされます。
- ファセットタグとバリアントタグの両方があるアクションは、ファセットとバリアントの両方でアクションのインストールが許可されている場合にインストールされます。

独自のファセットおよびバリアントタグを作成できます。Oracle Solaris では、次のタグが一般に使用されます。

バリアント名	取り得る値
variant.arch	sparc、i386
variant.opensolaris.zone	global、nonglobal
variant.debug.*	true、false

次のリストに、Oracle Solaris で使用される小さなファセットタグの例を示します。

```
facet.devel             facet.doc
facet.doc.html          facet.doc.info
facet.doc.man           facet.doc.pdf
facet.locale.de         facet.locale.en_GB
facet.locale.en_US      facet.locale.fr
facet.locale.ja_JP      facet.locale.zh_CN
```

イメージポリシー

イメージポリシーはブール値を持つイメージプロパティーによって定義されます。flush-content-cache-on-success および send-uuid プロパティーについてとそれらの値の表示および変更方法については、pkg(1) のマニュアルページのイメージプロパティーに関する項目を参照してください。

ファイル

pkg(5) イメージはより大きなファイルシステム内に任意に配置できるため、トークン $IMAGE_ROOT を使用して相対パスが区別されます。標準的なシステムインストールでは、$IMAGE_ROOT は / と同等です。

$IMAGE_ROOT/var/pkg
　完全または部分的なイメージのメタデータディレクトリ。

`$IMAGE_ROOT/.org.opensolaris,pkg`
ユーザーイメージのメタデータディレクトリ。

特定のイメージのメタデータ内の特定のファイルおよびディレクトリに、修復や復旧中に役立つ情報を含めることができます。トークン `$IMAGE_META` は、メタデータが含まれる最上位ディレクトリを参照するために使用されます。通常、`$IMAGE_META` は前述の 2 つのパスのいずれかです。

`$IMAGE_META/lost+found`
パッケージ操作中に移動された、競合するディレクトリおよびファイルの場所。

`$IMAGE_META/publisher`
パブリッシャーごとに 1 つのディレクトリが含まれます。各ディレクトリにはパブリッシャー固有のメタデータが格納されます。

`$IMAGE_META` ディレクトリ階層内のその他のパスは非公開であり、変更される可能性があります。

属性

次の属性については、`attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	`package/pkg`
インタフェースの安定性	不確実

関連項目

`pkg(1)`、`pkgsend(1)`、`pkg.depotd(1M)`、`pkg.sysrepo(1M)`、`svcs(1)`、`svcadm(1M)`

『Oracle Solaris 11.1 ソフトウェアパッケージの追加および更新』

『Oracle Solaris 11.1 パッケージリポジトリのコピーおよび作成』

『Oracle Solaris 11.1 での Image Packaging System を使用したソフトウェアのパッケージ化および配布』

`http://hub.opensolaris.org/bin/view/Project+pkg/`

名前

privileges – プロセスの特権モデル

機能説明

Oracle Solaris ソフトウェアには、プロセスのアクションをきめ細かく制御できるようにするための特権セットが実装されています。特定の特権の所有により、プロセスは、特定の一連の制限された操作を実行できるようになります。

開発者は、Oracle Solaris オペレーティングシステム内の主に特権に基づくセキュリティーモデルを変更することによって、プロセスをすべての特権 (スーパーユーザー) や特権なし (0 以外の UID) ではなく、実際に必要な特権操作に制限する機会が得られます。さらに、以前は制限されていなかった一連の操作にも特権が必要になりました。これらの特権は「基本」特権と呼ばれ、すべてのプロセスにデフォルトで付与されます。

まとめると、「基本」特権を除く定義されているすべての特権が、従来は root ユーザーに関連付けられた特権セットを構成します。「基本」特権は、以前、特権の設定されていないプロセスに与えられた「特権」です。

定義されている特権は次のとおりです:

`PRIV_CONTRACT_EVENT`
プロセスが、イベントの信頼できる配信をイベントエンドポイントにリクエストできるようにします。

プロセスが、ユーザーによって大量に生成される可能性のあるイベントをテンプレートのクリティカルイベント設定期間内に含めることができるようにします。

`PRIV_CONTRACT_IDENTITY`
プロセスがプロセス契約テンプレートのサービス FMRI 値を設定できるようにします。

`PRIV_CONTRACT_OBSERVER`
プロセスが、プロセスの実効ユーザー ID 以外のユーザーによって作成され、所有されている契約によって生成された契約イベントを監視できるようにします。

プロセスが、プロセスの実効ユーザー ID 以外のユーザーによって作成され、所有されている契約に属する契約イベントエンドポイントを開けるようにします。

`PRIV_CPC_CPU`
プロセスが CPU ごとのハードウェアパフォーマンスカウンタにアクセスできるようにします。

`PRIV_DTRACE_KERNEL`
DTrace のカーネルレベルのトレースを許可します。

`PRIV_DTRACE_PROC`
DTrace のプロセスレベルのトレースを許可します。プロセスレベルのトレースのプローブを、ユーザーがアクセス権を持っているプロセス内に配置し、有効にできるようにします。

PRIV_DTRACE_USER
DTrace のユーザーレベルのトレースを許可します。ユーザーがアクセス権を持っているプロセスを検査するために、システムコールおよびプロファイル DTrace プロバイダの使用を許可します。

PRIV_FILE_CHOWN
プロセスがファイルの所有者ユーザー ID を変更できるようにします。プロセスが、ファイルのグループ ID をプロセスの実効グループ ID 以外のグループ ID か、またはプロセスのいずれかの補助グループ ID に変更できるようにします。

PRIV_FILE_CHOWN_SELF
プロセスが自身のファイルを渡せるようにします。この特権を持つプロセスは、{_POSIX_CHOWN_RESTRICTED} が有効でないかのように動作します。

PRIV_FILE_DAC_EXECUTE
プロセスが、通常であればプロセスの実行権を禁止するアクセス権ビットまたは ACL を持つ実行可能ファイルを実行できるようにします。

PRIV_FILE_DAC_READ
プロセスが、通常であればプロセスの読み取り権を禁止するアクセス権ビットまたは ACL を持つファイルまたはディレクトリを読み取れるようにします。

PRIV_FILE_DAC_SEARCH
プロセスが、通常であればプロセスの検索権を許可しないアクセス権ビットまたは ACL を持つディレクトリを検索できるようにします。

PRIV_FILE_DAC_WRITE
プロセスが、プロセスの書き込み権を許可しないアクセス権ビットまたは ACL を持つファイルまたはディレクトリを書き込めるようにします。0 の実効 UID が存在しない場合に UID 0 によって所有されているファイルを書き込むには、すべての特権が必要です。

PRIV_FILE_DOWNGRADE_SL
プロセスが、ファイルまたはディレクトリの機密ラベルを既存の機密ラベルより優位でない機密ラベルに設定できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_FILE_FLAG_SET
プロセスが immutable、nounlink または appendonly ファイル属性を設定できるようにします。

PRIV_FILE_LINK_ANY
プロセスが、プロセスの実効 UID とは異なる UID によって所有されているファイルへのハードリンクを作成できるようにします。

PRIV_FILE_OWNER
ファイルの所有者でないプロセスが、そのファイルのアクセスや変更時間を変更できるようにします。ディレクトリの所有者でないプロセスが、そのディレクト

リのアクセスや変更時間を変更できるようにします。ファイルまたはディレクトリの所有者でないプロセスが、「実行後にテキストイメージを保存する」(スティッキー) ビットが設定された親ディレクトリを持つファイルまたはディレクトリの削除または名前変更を行うことができるようにします。ファイルの所有者でないプロセスが、そのファイル上に `namefs` をマウントできるようにします。ファイルまたはディレクトリの所有者でないプロセスが、そのファイルまたはディレクトリのアクセス権ビットまたは ACL を変更できるようにします。

`PRIV_FILE_READ`
プロセスが、プロセスの読み取り権を許可するアクセス権または ACL を持つファイルまたはディレクトリを読み取れるようにします。

`PRIV_FILE_SETID`
プロセスが、セットユーザー ID およびセットグループ ID ビットをクリアせずにファイルの所有権を変更したり、ファイルに書き込んだりできるようにします。プロセスが、プロセスの実効グループでも、プロセスのいずれかの補助グループでもないグループに属しているファイルまたはディレクトリにセットグループ ID ビットを設定できるようにします。プロセスが、`PRIV_FILE_OWNER` が存在する場合に、別の所有権を持つファイルにセットユーザー ID ビットを設定できるようにします。追加の制限は、`setuid 0` のファイルを作成または変更する場合に適用されます。

`PRIV_FILE_UPGRADE_SL`
プロセスが、ファイルまたはディレクトリの機密ラベルを既存の機密ラベルより優位な機密ラベルに設定できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_FILE_WRITE`
プロセスが、プロセスの書き込み権を許可するアクセス権または ACL を持つファイルまたはディレクトリを書き込めるようにします。

`PRIV_GRAPHICS_ACCESS`
プロセスがグラフィックスデバイスへの特権付き ioctl を作成できるようにします。通常、この特権を持つ必要があるのは xserver プロセスだけです。この特権を持つプロセスは、特権付きグラフィックスデバイスマッピングも実行できます。

`PRIV_GRAPHICS_MAP`
プロセスが、グラフィックスデバイスから特権付きマッピングを実行できるようにします。

`PRIV_IPC_DAC_READ`
プロセスが、通常であればプロセスの読み取り権を許可しないアクセス権ビットを持つ System V IPC メッセージキュー、セマフォーセット、または共有メモリーセグメントを読み取れるようにします。

PRIV_IPC_DAC_WRITE
プロセスが、通常であればプロセスの書き込み権を許可しないアクセス権ビットを持つ System V IPC メッセージキュー、セマフォーセット、または共有メモリーセグメントを書き込めるようにします。

PRIV_IPC_OWNER
System V IPC メッセージキュー、セマフォーセット、または共有メモリーセグメントの所有者でないプロセスが、メッセージキュー、セマフォーセット、または共有メモリーセグメントを削除したり、その所有権を変更したり、そのアクセス権ビットを変更したりできるようにします。

PRIV_NET_ACCESS
プロセスが TCP、UDP、SDP、または SCTP ネットワークエンドポイントを開けるようにします。

PRIV_NET_BINDMLP
プロセスが、プロセスのゾーンのマルチレベルポート (MLP) として構成されているポートにバインドできるようにします。この特権は、共有アドレスとゾーン固有のアドレスの両方の MLP に適用されます。MLP ポートの構成については、Trusted Extensions のマニュアルページの tnzonecfg(4) を参照してください。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_NET_ICMPACCESS
プロセスが ICMP パケットを送受信できるようにします。

PRIV_NET_MAC_AWARE
プロセスが、setpflags(2) を使用して NET_MAC_AWARE プロセスフラグを設定できるようにします。また、この特権により、プロセスは setsockopt(3SOCKET) を使用して SO_MAC_EXEMPT ソケットオプションも設定できるようになります。NET_MAC_AWARE プロセスフラグと SO_MAC_EXEMPT ソケットオプションはどちらも、ローカルプロセスのラベルがピアのデフォルトラベルより優位な場合や、ローカルプロセスが大域ゾーンで実行されている場合に、ローカルプロセスがラベルの付いていないピアと通信できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_NET_OBSERVABILITY
プロセスが、ネットワークトラフィックの受信やトラフィックの送信のみが禁止されているデバイスを開けるようにします。

PRIV_NET_PRIVADDR
プロセスが特権ポート番号にバインドできるようにします。特権ポート番号は 1 - 1023 (従来の UNIX 特権ポート) であり、さらに NFS や SMB での使用のために予約されているポートを除き、これらのポートは「udp/tcp_extra_priv_ports」とマークされています。

PRIV_NET_RAWACCESS
プロセスがネットワークレイヤーに直接アクセスできるようにします。

PRIV_PROC_AUDIT
プロセスが監査レコードを生成できるようにします。プロセスが独自の監査事前選択情報を取得できるようにします。

PRIV_PROC_CHROOT
プロセスが自身のルートディレクトリを変更できるようにします。

PRIV_PROC_CLOCK_HIGHRES
プロセスが高精度タイマーを使用できるようにします。

PRIV_PROC_EXEC
プロセスが exec(2) を呼び出せるようにします。

PRIV_PROC_FORK
プロセスが fork(2)、fork1(2)、または vfork(2) を呼び出せるようにします。

PRIV_PROC_INFO
プロセスが、シグナルを送信できるプロセス以外のプロセスのステータスを検査できるようにします。検査できないプロセスは /proc に表示できないため、存在していないように見えます。

PRIV_PROC_LOCK_MEMORY
プロセスが物理メモリー内のページをロックできるようにします。

PRIV_PROC_OWNER
プロセスが、所有権には関係なく、ほかのプロセスにシグナルを送信したり、ほかのプロセスにあるプロセスの状態を検査して変更したりできるようにします。別のプロセスを変更する場合は、追加の制限が適用されます: 接続しているプロセスの実効特権セットは、ターゲットプロセスの実効セット、許可セット、および継承可能セットのスーパーセットである必要があります。制限セットは、ターゲットの制限セットのスーパーセットである必要があります。ターゲットプロセスでいずれかの UID が 0 に設定されている場合は、実効 UID が 0 でないかぎり、すべての特権が表明されている必要があります。プロセスが CPU に任意のプロセスをバインドできるようにします。

PRIV_PROC_PRIOCNTL
プロセスが、自身の優先順位を現在のレベルを超えて引き上げることができるようにします。プロセスが、自身のスケジューリングクラスを任意のスケジューリングクラス (RT クラスを含む) に変更できるようにします。

PRIV_PROC_SESSION
プロセスが、シグナルまたはトレースプロセスをセッションの外部に送信できるようにします。

PRIV_PROC_SETID
UID 0 ですべての特権が表明されている必要がある場合、プロセスが自身の UID を自由に設定できるようにします。

PRIV_PROC_TASKID
プロセスが、呼び出し元プロセスに新しいタスク ID を割り当てることができるようにします。

PRIV_PROC_ZONE
プロセスがシグナルをトレースしたり、ほかのゾーン内のプロセスに送信したりできるようにします。zones(5) を参照してください。

PRIV_SYS_ACCT
プロセスが、acct(2) によるアカウンティングを有効または無効にしたり、管理したりできるようにします。

PRIV_SYS_ADMIN
プロセスが、ノードおよびドメイン名の設定や、coreadm(1M) および nscd(1M) 設定の指定などのシステム管理タスクを実行できるようにします。

PRIV_SYS_AUDIT
プロセスが (カーネル) 監査デーモンを起動できるようにします。プロセスが監査状態 (監査ユーザー ID、監査端末 ID、監査セッション ID、監査事前選択マスク) を表示したり、設定したりできるようにします。プロセスが監査を無効にしたり、有効にしたりできるようにします。プロセスが監査パラメータ (キャッシュとキューのサイズ、イベントからクラスへのマッピング、およびポリシーオプション) を構成できるようにします。

PRIV_SYS_CONFIG
プロセスがさまざまなシステム構成タスクを実行できるようにします。ファイルシステム固有の管理手順 (ファイルシステム構成 ioctl、割り当て制限の呼び出し、スナップショットの作成と削除、PCFS ブートセクターの操作など) を許可します。

PRIV_SYS_DEVICES
プロセスがデバイス特殊ファイルを作成できるようにします。プロセスが、許可されたアクセスをチェックするために、カーネルの drv_priv(9F) 関数を呼び出すカーネルモジュールを正常に呼び出せるようにします。プロセスが、実際のコンソールデバイスを直接開けるようにします。プロセスが、排他的に開かれていたデバイスを開けるようにします。

PRIV_SYS_DL_CONFIG
プロセスがシステムのデータリンクインタフェースを構成できるようにします。

PRIV_SYS_IP_CONFIG
プロセスがシステムの IP インタフェースと経路を構成できるようにします。プロセスが TCP/IP パラメータを構成できるようにします。プロセスが、アンカーされた STREAMS モジュールを一致する zoneid とともに pop できるようにします。

PRIV_SYS_IPC_CONFIG
プロセスが System V IPC メッセージキューバッファーのサイズを大きくすることができるようにします。

PRIV_SYS_LINKDIR
プロセスがディレクトリのリンクを解除したり、リンクしたりできるようにします。

PRIV_SYS_MOUNT
プロセスが、通常であれば制限されるファイルシステム (つまり、namefs を除くほとんどのファイルシステム) をマウントしたり、マウント解除したりできるようにします。プロセスがスワップデバイスを追加したり、削除したりできるようにします。

PRIV_SYS_NET_CONFIG
プロセスが、PRIV_SYS_IP_CONFIG、PRIV_SYS_DL_CONFIG、および PRIV_SYS_PPP_CONFIG で許可されるすべての操作に加えて、次の操作を実行できるようにします: rpcmod STREAMS モジュールの使用、モジュールのスタックの先頭以外の場所にある STREAMS モジュールの挿入および削除。

PRIV_SYS_NFS
プロセスが NFS サービスを行うことができるようにします: NFS カーネルスレッドの起動、NFS ロック処理の実行、NFS で予約されたポート (ポート 2049 (nfs) およびポート 4045 (lockd)) へのバインド。

PRIV_SYS_PPP_CONFIG
プロセスが pppd(1M) pppd(1M) で PPP インスタンスを作成、構成、および破棄したり、sppptun(1M)sppptun(1M) で PPPoE の plumb を制御したりできるようにします。この特権は、排他的な IP スタックインスタンスゾーンにデフォルトで付与されます。

PRIV_SYS_RES_BIND
プロセスがプロセッサセットにプロセスをバインドできるようにします。

PRIV_SYS_RES_CONFIG
プロセスが、この段落で概要が説明されている次の操作に加えて、プロセッサセットにプロセスを PRIV_SYS_RES_BIND としてバインドできるようにします。プロセスがプロセッサセットを作成および削除したり、プロセッサセットに CPU を割り当てたり、PSET_NOESCAPE プロパティーをオーバーライドしたりできるようにします。プロセスが、p_online(2) を使用してシステム内の CPU の動作ステータスを変更できるようにします。プロセスがファイルシステムの割り当て制限を構成できるようにします。プロセスがリソースプールを構成したり、プールにプロセスをバインドしたりできるようにします。

PRIV_SYS_RESOURCE
プロセスが、setrlimit(2) および setrctl(2) を使用して、自身に課せられたリソース制限を超えることができるようにします。

PRIV_SYS_SHARE
プロセスがファイルシステムを共有したり、共有解除したりできるようにします。

PRIV_SYS_SMB
プロセスが NetBIOS または SMB サービスを行うことができるようにします: SMB カーネルスレッドの起動、NetBIOS または SMB で予約されたポート (ポート 137、138、139 (NetBIOS)、および 445 (SMB)) へのバインド。

PRIV_SYS_SUSER_COMPAT
プロセスが、許可されたアクセスをチェックするために、カーネルの suser() 関数を呼び出すサードパーティー製のロード可能なモジュールを正常に呼び出せるようにします。この特権は、サードパーティー製のロード可能なモジュールの互換性のためにのみ存在し、Oracle Solaris では使用されません。

PRIV_SYS_TIME
プロセスが、次のうちのいずれかの適切なシステムコールを使用してシステム時間を操作できるようにします: stime(2)、adjtime(2)、ntp_adjtime(2)。

PRIV_SYS_TRANS_LABEL
プロセスが、プロセスの機密ラベルと同等か、またはより優位なラベルを外部の文字列形式との間で変換できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_VIRT_MANAGE
プロセスが仮想化環境を管理できるようにします。

PRIV_WIN_COLORMAP
プロセスがカラーマップの制限をオーバーライドできるようにします。

プロセスがカラーマップをインストールまたは削除できるようにします。

プロセスが、ほかのプロセスによって割り当てられたカラーマップセルエントリを取得できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_CONFIG
プロセスが、X サーバーによって永続的に保持されているリソースを構成または破棄できるようにします。

プロセスがスクリーンセーバーのタイムアウト値を設定するために SetScreenSaver を使用できるようにします。

プロセスが表示アクセス制御リストを変更するために ChangeHosts を使用できるようにします。

プロセスが GrabServer を使用できるようにします。

プロセスが、ウィンドウ、ピクセルマップ、カラーマップ、プロパティー、カーソル、フォント、またはグラフィックコンテキストのリソースを保持できる SetCloseDownMode リクエストを使用できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_WIN_DAC_READ`
プロセスが、所有していない (別のユーザー ID を持つ) ウィンドウリソースから読み取れるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_WIN_DAC_WRITE`
プロセスが、所有していない (別のユーザー ID を持つ) ウィンドウリソースに書き込んだり、このようなウィンドウリソースを作成したりできるようにします。新しく作成されるウィンドウのプロパティーは、そのウィンドウのユーザー ID を使用して作成されます。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_WIN_DEVICES`
プロセスがウィンドウ入力デバイスに対する操作を実行できるようにします。

プロセスがキーボードやポインタの制御を取得したり、設定したりできるようにします。

プロセスがポインタボタンやキーのマッピングを変更できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_WIN_DGA`
プロセスがダイレクトグラフィックスアクセス (DGA) X プロトコルの拡張機能を使用できるようにします。フレームバッファーへの直接のプロセスアクセスは引き続き必要です。このため、プロセスがフレームバッファーへのアクセスを許可する MAC および DAC 特権を持っているか、またはそのプロセスにフレームバッファーが割り当てられている必要があります。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

`PRIV_WIN_DOWNGRADE_SL`
プロセスが、ウィンドウリソースの機密ラベルを既存の機密ラベルより優位でない機密ラベルに設定できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_FONTPATH
プロセスがフォントパスを設定できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_MAC_READ
プロセスが、プロセスの機密ラベルと等しくない機密ラベルを持つウィンドウリソースから読み取れるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_MAC_WRITE
プロセスが、プロセスの機密ラベルと等しくない機密ラベルを持つウィンドウリソースを作成できるようにします。新しく作成されるウィンドウのプロパティーは、そのウィンドウの機密ラベルを使用して作成されます。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_SELECTION
プロセスが、選択確認ダイアログボックスの介入なしでウィンドウ間のデータ移動をリクエストできるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

PRIV_WIN_UPGRADE_SL
プロセスが、ウィンドウリソースの機密ラベルを既存の機密ラベルより優位な機密ラベルに設定できるようにします。

この特権は、システムが Trusted Extensions で構成されている場合にのみ解釈されます。

上のに示された特権のうち、PRIV_FILE_LINK_ANY、PRIV_FILE_READ、PRIV_FILE_WRITE、PRIV_PROC_INFO、PRIV_PROC_SESSIO
の各特権が「基本」特権と見なされます。これらは、特権の設定されていないプロセスから常に使用可能であった特権です。デフォルトでは、プロセスには引き続き基本特権が割り当てられます。

setuid root exec を成功させる (つまり、0 の実効 UID と追加の特権を取得する) には、プロセスの制限セット (次を参照) 内に
PRIV_PROC_SETID、PRIV_PROC_AUDIT、PRIV_SYS_RESOURCE の各特権が存在する必要があります。

Oracle Solaris での特権の実装では、プロセス資格が次の 4 つの特権セットで拡張されています:

I、継承可能セット	exec で継承される特権。
P、許可セット	プロセスのための最大の特権セット。
E、実効セット	現在有効な特権。
L、制限セット	プロセスおよびその子孫が取得できる特権の上限。L への変更は、次回の exec で有効になります。

I、P、E の各セットは通常、特権の設定されていないプロセスの基本的な特権セットと同じです。制限セットは通常、完全な特権セットです。

各プロセスには、PA (特権を認識する) と NPA (PA 以外) の値を取ることができる特権認識状態 (PAS) があります。PAS は、古いスーパーユーザーモデルとの完全な互換性と、実効 UID の完全な無視のどちらかを選択できるようにするための移行メカニズムです。

説明を簡単にするために、「監視された実効セット」(oE) および「監視された許可セット」(oP) の概念と、実装セット iE および iP を導入します。

プロセスは、setppriv(2) によって実効セット、許可セット、制限特権セットを操作するか、または setpflags(2) を使用することによって、特権を認識するようになります。いずれの場合も、特権を認識するようになるプロセスでは oE および oP は不変です。特権を認識するようになるプロセスでは、次の割り当てが実行されます:

```
iE = oE
iP = oP
```

プロセスが特権を認識する場合、UID の変更の下で oE および oP は不変です。プロセスが特権を認識しない場合、oE および oP は次のように監視されます:

```
oE = euid == 0 ? L : iE
oP = (euid == 0 || ruid == 0 || suid == 0) ? L : iP
```

特権を認識しないプロセスに 0 の実効 UID が割り当てられている場合、そのプロセスは、その制限セット (特権の上限) に含まれている特権を行使できます。特権を認識しないプロセスにいずれかの UID 0 が割り当てられている場合は、潜在的に L 内のすべての特権を行使できることを示します。

setpflags() を使用すると、特権を認識しない状態にプロセスを戻すことができます。カーネルは、exec(2) で、常にこれを試みます。この操作は、次の条件が満たされた場合にのみ許可されます:

- いずれかの UID が 0 に等しい場合は、P が L に等しいことが必要です。
- 実効 UID が 0 に等しい場合は、E が L に等しいことが必要です。

プロセスが特権の認識を放棄した場合は、次の割り当てが実行されます:

```
if (euid == 0) iE = L & I
if (any uid == 0) iP = L & I
```

0 の UID を持っていないときに取得された特権は、制限セットで制限された、プロセスの継承可能セットです。

プロセスの (監視された) 実効特権セット内の特権によってのみ、プロセスは制限された操作の実行を許可されます。プロセスは、いずれかの特権操作関数を使用して特権セットに特権を追加したり、特権セットから特権を削除したりすることができます。特権は、常に削除できます。実効セットや継承可能セットに追加できるのは、許可セットにある特権だけです。制限セットは拡張できません。継承可能セットは、許可セットより大きくすることができます。

プロセスが exec(2) を実行すると、カーネルは次の特権セットの変更を行う前に、まず特権の認識を放棄しようとします:

```
E' = P' = I' = L & I
L is unchanged
```

プロセスが自身の特権を操作していない場合は、E、P、および I がすでに同一であるため、特権セットは実質的に同じままになります。

制限セットは、exec の実行時に適用されます。

特権を認識しないアプリケーションを下位互換性がある方法で実行するには、特権を認識するアプリケーションが、特権を認識しないアプリケーションを I=basic で起動するようにしてください。

ほとんどの特権の場合は、その特権が存在しなければ単に失敗するだけです。場合によっては、特権が存在しないためにシステムコールが異なった動作をすることがあります。その他の場合として、特権を削除すると、強制的に setuid ルートアプリケーションの重大な誤動作が引き起こされることがあります。このタイプの特権は「安全ではない」と見なされます。プロセスの制限セットに、いずれかの安全ではない特権が含まれていない場合、システムは setuid ルートアプリケーションの setuid ビットに従いません。実効ユーザー ID を変更することも、特権を増やすこともなく、setuid ルートアプリケーションの実行が続行されます。次の安全ではない特権が識別されています: PRIV_PROC_SETID、PRIV_SYS_RESOURCE、および PRIV_PROC_AUDIT。

特権エスカレーション

特定の状況では、1 つの特権によってプロセスが、そのプロセスに明示的には付与されていなかった 1 つ以上の追加の特権を取得する可能性があります。特権のこのようなエスカレーションを回避するために、セキュリティーポリシーには、これらの追加の特権に対する明示的なアクセス権が必要になります。

エスカレーションの一般的な例として、「生の」インタフェースによるシステムリソースの変更を許可するメカニズムがあります。たとえば、/dev/kmem によるカーネルデータ構造の変更や、/dev/dsk/* によるファイルの変更があります。エス

カレーションはまた、プロセスが、制御プロセスより多くの特権を使用してプロセスを制御する場合にも発生します。この特殊な場合として、UID 0 によって所有されているオブジェクトの操作または作成や、setuid(2) を使用して UID 0 を取得しようとすることがあります。UID 0 はすべてのシステム構成ファイルを所有するだけでなく、通常のファイル保護メカニズムでは UID 0 を持つプロセスがシステム構成を変更できるため、UID 0 には特別な扱いが必要です。適切なファイル変更を使用すると、0 の実効 UID で実行されている特定のプロセスがすべての特権を取得できます。

プロセスが UID 0 を取得する可能性のある状況では、セキュリティーポリシーに追加の特権 (最大の場合、完全な特権セット) が必要です。システムファイルの保護のための追加のメカニズムが使用できる場合には、このような制限を緩和または解消できる可能性があります。Oracle Solaris の現在のリリースには、このようなメカニズムはありません。

UID 0 プロセスの使用は、できるだけ制限するようにしてください。別の UID の下で実行されていて、しかも必要な特権を持つプログラムに置き換えるようにしてください。

サブプロセスを exec で実行する必要がないデーモンの場合は、その許可セットと制限セットから PRIV_PROC_EXEC 特権を削除するようにしてください。

割り当てられる特権と保護手段

ユーザーに特権が割り当てられる場合、システム管理者はそのユーザーに、意図したより多くの権限を与える可能性があります。管理者は、保護手段が必要かどうかを考慮するべきです。たとえば、ユーザーに PRIV_PROC_LOCK_MEMORY 特権が与えられる場合、そのユーザーがすべてのメモリーをロックしないようにするために、管理者は project.max-locked-memory リソース制御も設定することを考慮するべきです。

拡張ポリシー

構成ファイル内またはコマンド行に特権が一覧表示されているときに、拡張ポリシーの構文を使用できる場合があります。拡張ポリシーとは、特権セットを中括弧で囲み、そのあとにコロンとオブジェクトを続けたものです。拡張ポリシーによって、指定されたオブジェクトに対してその特権の一覧を使用する機能が追加されます。現在は、ネットワークポート、UID、およびファイルオブジェクトのための拡張ポリシーがサポートされています。

たとえば、

`{file_dac_read}:/var/core/*`
: /var/core の下にあるファイルへのアクセス時に特権 file_dac_read を使用できるようにします。

`{net_privaddr}:80/tcp,{net_privaddr}:443/tcp`
: プロセスが TCP ポート 80 および 443 にネットワークエンドポイントをバインドできるようにします。

`{proc_setid}:80-100`
プロセスが UID を UID 80 から 100 まで (80 と 100 を含む) に変更できるようにします。

`{proc_setid}:casper`
プロセスが UID をユーザー名「casper」の UID に変更できるようにします。

拡張ポリシーに一覧表示されている特権セットは、そのポリシーの特権セットにゾーン内で使用可能なすべての特権が含まれていないかぎり、その拡張ポリシーがインストールされたときに継承可能セットから削除され、その結果、許可セットと実効セットから削除されます。たとえば、`{zone}:/etc/shadow` という形式の拡張ポリシーをインストールした場合は、どの特権も削除されません。これに対して、拡張ポリシーの形式が `{file_dac_read}:/etc/shadow` である場合は、`PRIV_FILE_DAC_READ` 特権が削除されます。

拡張ポリシーは、実効セットに特権が存在しない場合にのみ有効になります。

`{all}:/some/file` などの拡張ポリシーを指定することは可能ですが、システムは引き続き、`setuid` ビットの追加など、一部のアプリケーションを制限します。

拡張ポリシーに一覧表示されているすべての特権は、プロセスがそのポリシーをインストールするときに、そのプロセス内で有効である必要があります。たとえば、ポリシーをインストールするために ppriv(1) が使用される場合は、その実効セットにすべての特権が一覧表示されている必要があります。`ppriv` のオブジェクトであるプロセスには、このような制限は適用されません。ただし、その制限セットによって拡張ポリシー内のすべての特権がオーバーライドされます。

一部のコンテキストでは、拡張ポリシーの構文の一部をエスケープする必要があります。たとえば、exec_attr(4) では、コロン (:) をバックスラッシュ (\) でエスケープする必要があります。また、コンテキストによっては、ファイル名にある一部の文字もバックスラッシュでのエスケープが必要になる可能性があります。

拡張ポリシーは、ファイルシステム内のすべてのレイヤーで評価されます。lofs(7FS) ファイルシステムの場合、指定されたポリシーでは、そのポリシーが `lofs` ファイルシステムと基礎となるファイルシステムの両方を指定する必要があるという点を考慮に入れる必要があります。

次の一覧には、オブジェクトとそれに関連する特権のタイプが含まれています。

オブジェクト	構文	特権
ユーザー名	名前	`proc_setid`
Uid	uid	`proc_setid`
uid の範囲	uid1-uid2	`proc_setid`

オブジェクト	構文	特権
ネットワークポート[1]	port/udp、port/tcp、 port/sctp、port/*	net_privaddr
ポートの範囲	port1-port2/<proto>	net_privaddr
ファイル名	パス名	ファイル特権、proc_exec
ワイルドカード[2]	パス名*	ファイル特権、proc_exec

[1] services(4) で定義された数値

[2] 指定されたパス名で始まるすべてのファイル名に一致する

特権のデバッグ

システムコールがアクセス権のエラーで失敗した場合は、その問題の原因がただちに明らかになるとは限りません。このような問題をデバッグするには、特権のデバッグと呼ばれるツールを使用できます。あるプロセスに対して特権のデバッグが有効になっている場合、カーネルは、存在しない特権をそのプロセスの制御端末に報告します。(プロセスに対するデバッグは、ppriv(1) の -D オプションを使用して有効にします。)さらに、管理者は次を使用して system(4) の変数 priv_debug を設定することにより、システム規模の特権のデバッグを有効にすることができます:

```
set priv_debug = 1
```

実行中のシステムでは、mdb(1) を使用すると、この変数を変更できます。

特権の管理

ユーザーまたは役割に特権を割り当てたり、ユーザーまたは役割への特権を変更したりするには、それぞれ、usermod(1M) または rolemod(1M) を使用します。システム上でサポートされている特権を列挙するには ppriv(1) を、プログラムにどの特権が必要かを確認するには truss(1) を使用します。

関連項目

mdb(1)、ppriv(1)、add_drv(1M)、ifconfig(1M)、lockd(1M)、nfsd(1M)、pppd(1M)、rem_drv
setegid(2)、seteuid(2)、setgid(2)、setgroups(2)、setpflags(2)、setppriv(2)、setrctl(2)、
t_bind(3NSL)、timer_create(3C)、ucred_get(3C)、exec_attr(4)、proc(4)、
services(4)、system(4)、user_attr(4)、
lofs(7FS)ddi_cred(9F)、drv_priv(9F)、priv_getbyname(9F)、priv_policy(9F)、priv_policy_

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

注意事項

プロセスからいずれかの基本特権を削除することは、そのプロセスを標準に準拠しない状態のままにするだけでなく、予期しないアプリケーション障害を引き起こす可能性があるため、その潜在的な副作用を完全に理解した上でのみ行うようにしてください。

名前

prof – profile within a function

形式

```
#define MARK
#include <prof.h>

void MARK(name);
```

機能説明

MARK introduces a mark called *name* that is treated the same as a function entry point. Execution of the mark adds to a counter for that mark, and program-counter time spent is accounted to the immediately preceding mark or to the function if there are no preceding marks within the active function.

name may be any combination of letters, numbers, or underscores. Each *name* in a single compilation must be unique, but may be the same as any ordinary program symbol.

For marks to be effective, the symbol MARK must be defined before the header prof.h is included, either by a preprocessor directive as in the synopsis, or by a command line argument:

```
cc –p –DMARK work.c
```

If MARK is not defined, the MARK(*name*) statements may be left in the source files containing them and are ignored. prof –g must be used to get information on all labels.

使用例

In this example, marks can be used to determine how much time is spent in each loop. Unless this example is compiled with MARK defined on the command line, the marks are ignored.

```
#include <prof.h>
work( )
{
        int i, j;
        . . .
        MARK(loop1);
        for (i = 0; i < 2000; i++) {
                . . .
        }
        MARK(loop2);
        for (j = 0; j < 2000; j++) {
                . . .
        }
}
```

関連項目

profil(2), monitor(3C)

名前

rbac, RBAC – 役割に基づくアクセス制御

機能説明

役割に基づくアクセス制御を使用すると、システム管理者は、システムの各部分の管理制御をユーザーに委任できます。ユーザーが追加の特権を使用してコマンドを実行できるようにするための方法には、次の 2 つがあります:

- ユーザーにプロファイルを直接割り当てます。この場合、追加の認証は必要ありません
- 役割を作成し、その役割にプロファイルを割り当てます。また、このアクセス制御を使用すると、通常であれば実行を許可されるコマンドを実行できなくすることにより、ユーザーにとって制限的な環境を構築することもできます。

プロファイル

プロファイルは、追加の特権または特定の実 UID/GID や実効 UID/GID、あるいはその両方で実行されるコマンドおよび承認の名前付き集合です。たとえば、ほとんどのプリンタシステムは、`lp` コマンドを UID または `lp` で実行することによって管理できます。一部のコマンドの実行には、`privileges(5)` で定義された特権が必要です。たとえば、「Process Management」プロファイルを使用すると、ユーザーは、所有していないプロセスにシグナルを送信できるように `proc_owner` 特権で `kill` コマンドを実行できます。

管理者がシステムで提供されたプロファイルを拡張したり、独自のプロファイルを作成したりする方法については、`exec_attr(4)` および `prof_attr(4)` を参照してください。プロファイルの構成は、現在サポートされているネームサービス (ファイル、NIS、LDAP) のいずれかに格納できます。

プロファイルはまた、サービスの実行に使用される特権や UID/GID を制御するためにサービス管理機能 (SMF) でも使用できます。詳細は、`smf_security(5)` を参照してください。

役割

役割は、承認ユーザーが `su(1M)` コマンドで、またはホストベース認証や GSS-API 認証を使用している場合は `ssh(1)` でネットワーク経由でしかアクセスできないシステムに直接ログインすることができない特殊な共有アカウントです。役割は、`rlogin(1)`、`telnet(1)`、または `gdm` ではログインできません。

役割には、通常ユーザーと同様に UID、パスワード、およびホームディレクトリがあります。役割に対する認証は、ユーザー独自のパスワードまたは役割ごとのパスワードのどちらかで行うことができます (`user_attr(4)` の `roleauth` キーワードによって、その動作が役割ごとに制御されます)。通常、役割のログインシェルは、1 つ以上のプロファイルが付与されることによって役割が常に特権でコマンドを実行できるようにするプロファイルシェル (`pfsh(1)`、`pfksh(1)`、`pfcsh(1)`) のいずれかです。

役割は通常、共有アカウント環境が必要な場合にのみ必要になります。通常は、ユーザーにプロファイルを直接割り当てれば十分です。

root ユーザーは、`usermod(1M)` コマンドを使用すると役割として構成できます。これにより、root パスワードがより広範囲に知られている場合でも、承認ユーザーのみが root になれることが保証されます。

```
# usermod -K type=role root
```

root を役割にしても、シングルユーザーモードへのアクセスは制限されません。システムコンソールは、eeprom(1M) でのセキュリティーパスワードの設定などの、ほかの手段を使用して保護するようにしてください。

承認

承認は、一部の操作または操作のクラスを実行するためのユーザーの権利を表す一意の文字列です。承認は通常、常に何らかの特権で実行されるプログラム (たとえば、cdrw(1) などの setuid(2) プログラムや、システムの cron(1M) デーモン) によってのみチェックされます。

承認の定義は、auth_attr(4) データベースに格納されます。プログラミングでの承認チェックの場合は、承認名のみが重要です。

auth_attr データベース内のいくつかの代表的な値を次に示します:

```
solaris.jobs.:::Cron and At Jobs::help=JobHeader.html
solaris.jobs.grant:::Delegate Cron & At \
    Administration::help=JobsGrant.html
solaris.jobs.admin:::Manage All Jobs::help=AuthJobsAdmin.html
solaris.jobs.user:::Cron & At User::help=JobsUser.html
```

grant 接尾辞で終わる承認名文字列は、ユーザーが、同じ接頭辞と機能領域を持つ承認をほかのユーザーに委任できるようにする特殊な承認です。

solaris で始まる承認名はすべて、オペレーティングシステムのベンダーによる割り当てのために予約されています。開発者と管理者は、独自の最上位の名前空間を作成できます (会社名、DNS ドメイン名、アプリケーション名などの一意の識別子を使用することをお勧めします)。

承認のチェック

C コードから承認をチェックするには、開発者は、ユーザーに特定の承認が割り当てられているかどうかを検証する chkauthattr(3C) ライブラリ関数を使用するようにしてください。

承認は、auths(1) ユーティリティーの出力をチェックすることによって、シェルスクリプトで明示的にチェックできます。たとえば、

```
for auth in ‘auths      | tr , " "‘ NOTFOUND
do
    ["$auth" = "solaris.date" ] && break      # authorization found
done

if [ "$auth" != "solaris.date" ]
then
    echo >&2 "$PROG: ERROR: you are not authorized to set the date"
    exit 1
fi
```

承認はまた、どのユーザーがサービスの状態を変更したり、サービスを再構成したりできるかを制御するためにサービス管理機能 (SMF) でも使用されます。詳細は、smf_security(5) を参照してください。

sudo(1M) との比較

Solaris の RBAC には、UNIX または UNIX ライクなシステムには多くの場合、用意されている sudo(1M) に似た一連の機能が用意されています。これは Solaris の Companion CD で提供されています。

Solaris RBAC と sudo のもっとも明らかな相違点の 1 つが認証モデルです。sudo では、ユーザーは自分自身として再認証します。Solaris RBAC では、追加の認証が必要ないか (ユーザーにプロファイルが直接割り当てられている場合)、またはユーザーが、役割と呼ばれる共有アカウントから認証されるかのどちらかです。

sudo での NOPASSWD 機能の使用は、ユーザーにプロファイルを割り当て、ユーザーが pfexec(1) を使用してコマンドを実行するようにする方法と同様です。たとえば、sudoers(4) で、ユーザーが UID 0 として kill(1) を認証なしで (NOPASSWD) 実行できるようにする場合、ユーザーは次を実行します:

```
$ sudo kill -HUP 1235
```

Solaris RBAC では、ユーザーに通常の (つまり、プロファイルなしの) ログインシェルが割り当てられている場合、ユーザーは「Process Management」プロファイルが割り当てられることにより同等の操作を実行し、次のように pfexec を使用します:

```
$ pfexec kill -HUP 1235
```

ユーザーにログインシェルとして (pfsh などの) プロファイルシェルが割り当てられている場合は、「接頭辞」がなくても kill は常に追加の特権で実行されます。たとえば、

```
$ kill -HUP 1235
```

RBAC の役割は、ユーザーパスワードではなく役割のパスワードが必要な点を除き、概念的には sudoers(4) の User_Alias に似ています。

RBAC の実行プロファイル (exec_attr(4) エントリ) は、sudoers の Cmnd_Alias に似ています。

現在、Solaris RBAC には Host_Alias sudo(1M) と同等の機能はありません。

関連項目

auths(1)、ld.so.1(1)、pfcsh(1)、pfexec(1)、pfksh(1)、pfsh(1)、roles(1)、sudo(1M)、exec_

『Oracle Solaris 11.1 の管理: セキュリティーサービス』

名前

regex – internationalized basic and extended regular expression matching

機能説明

Regular Expressions (REs) provide a mechanism to select specific strings from a set of character strings. The Internationalized Regular Expressions described below differ from the Simple Regular Expressions described on the regexp(5) manual page in the following ways:

- both Basic and Extended Regular Expressions are supported
- the Internationalization features—character class, equivalence class, and multi-character collation—are supported.

The Basic Regular Expression (BRE) notation and construction rules described in the BASIC REGULAR EXPRESSIONS section apply to most utilities supporting regular expressions. Some utilities, instead, support the Extended Regular Expressions (ERE) described in the EXTENDED REGULAR EXPRESSIONS section; any exceptions for both cases are noted in the descriptions of the specific utilities using regular expressions. Both BREs and EREs are supported by the Regular Expression Matching interfaces regcomp(3C) and regexec(3C).

Basic Regular Expressions

BREs Matching a Single Character

A BRE ordinary character, a special character preceded by a backslash, or a period matches a single character. A bracket expression matches a single character or a single collating element. See RE Bracket Expression, below.

BRE Ordinary Characters

An ordinary character is a BRE that matches itself: any character in the supported character set, except for the BRE special characters listed in BRE Special Characters, below.

The interpretation of an ordinary character preceded by a backslash (\) is undefined, except for:

1. the characters), (, {, and }
2. the digits 1 to 9 inclusive (see BREs Matching Multiple Characters, below)
3. a character inside a bracket expression.

BRE Special Characters

A BRE *special character* has special properties in certain contexts. Outside those contexts, or when preceded by a backslash, such a character will be a BRE that matches the special character itself. The BRE special characters and the contexts in which they have their special meaning are:

. [\ The period, left-bracket, and backslash are special except when used in a bracket expression (see RE Bracket Expression, below). An expression containing a [that is not preceded by a backslash and is not part of a bracket expression produces undefined results.

* The asterisk is special except when used:

- in a bracket expression
- as the first character of an entire BRE (after an initial ^, if any)

- as the first character of a subexpression (after an initial ^, if any); see BREs Matching Multiple Characters, below.

^ The circumflex is special when used:

- as an anchor (see BRE Expression Anchoring, below).
- as the first character of a bracket expression (see RE Bracket Expression, below).

$ The dollar sign is special when used as an anchor.

Periods in BREs

A period (.), when used outside a bracket expression, is a BRE that matches any character in the supported character set except NUL.

RE Bracket Expression

A bracket expression (an expression enclosed in square brackets, [ ]) is an RE that matches a single collating element contained in the non-empty set of collating elements represented by the bracket expression.

The following rules and definitions apply to bracket expressions:

1. A *bracket expression* is either a matching list expression or a non-matching list expression. It consists of one or more expressions: collating elements, collating symbols, equivalence classes, character classes, or range expressions (see rule 7 below). Portable applications must not use range expressions, even though all implementations support them. The right-bracket (]) loses its special meaning and represents itself in a bracket expression if it occurs first in the list (after an initial circumflex (^), if any). Otherwise, it terminates the bracket expression, unless it appears in a collating symbol (such as [.].]) or is the ending right-bracket for a collating symbol, equivalence class, or character class. The special characters:

   ```
   .   *   [   \
   ```

   (period, asterisk, left-bracket and backslash, respectively) lose their special meaning within a bracket expression.

   The character sequences:

   ```
   [.   [=   [:
   ```

   (left-bracket followed by a period, equals-sign, or colon) are special inside a bracket expression and are used to delimit collating symbols, equivalence class expressions, and character class expressions. These symbols must be followed by a valid expression and the matching terminating sequence .], =] or :], as described in the following items.
2. A *matching list* expression specifies a list that matches any one of the expressions represented in the list. The first character in the list must not be the circumflex. For example, `[abc]` is an RE that matches any of the characters `a`, `b` or `c`.
3. A *non-matching list* expression begins with a circumflex (^), and specifies a list that matches any character or collating element except for the expressions represented in the list after the leading circumflex. For example, [^abc] is an RE that matches any character or

collating element except the characters `a`, `b`, or `c`. The circumflex will have this special meaning only when it occurs first in the list, immediately following the left-bracket.

4. A *collating symbol* is a collating element enclosed within bracket-period ([..]) delimiters. Multi-character collating elements must be represented as collating symbols when it is necessary to distinguish them from a list of the individual characters that make up the multi-character collating element. For example, if the string `ch` is a collating element in the current collation sequence with the associated collating symbol <ch>, the expression [[.ch.]] will be treated as an RE matching the character sequence `ch`, while [ch] will be treated as an RE matching `c` or `h`. Collating symbols will be recognized only inside bracket expressions. This implies that the RE `[[.ch.]]*c` matches the first to fifth character in the string chchch. If the string is not a collating element in the current collating sequence definition, or if the collating element has no characters associated with it, the symbol will be treated as an invalid expression.

5. An *equivalence class expression* represents the set of collating elements belonging to an equivalence class. Only primary equivalence classes will be recognised. The class is expressed by enclosing any one of the collating elements in the equivalence class within bracket-equal ([==]) delimiters. For example, if a and b belong to the same equivalence class, then [[=a=]b], [[==]b] and [[==]b] will each be equivalent to [ab]. If the collating element does not belong to an equivalence class, the equivalence class expression will be treated as a *collating symbol*.

6. A *character class expression* represents the set of characters belonging to a character class, as defined in the `LC_CTYPE` category in the current locale. All character classes specified in the current locale will be recognized. A character class expression is expressed as a character class name enclosed within bracket-colon ([::]) delimiters.

   The following character class expressions are supported in all locales:

`[:alnum:]`	`[:cntrl:]`	`[:lower:]`	`[:space:]`
`[:alpha:]`	`[:digit:]`	`[:print:]`	`[:upper:]`
`[:blank:]`	`[:graph:]`	`[:punct:]`	`[:xdigit:]`

   In addition, character class expressions of the form:

   ```
   [:name:]
   ```

   are recognized in those locales where the *name* keyword has been given a `charclass` definition in the `LC_CTYPE` category.

7. A *range expression* represents the set of collating elements that fall between two elements in the current collation sequence, inclusively. It is expressed as the starting point and the ending point separated by a hyphen (–).

Range expressions must not be used in portable applications because their behavior is dependent on the collating sequence. Ranges will be treated according to the current collating sequence, and include such characters that fall within the range based on that collating sequence, regardless of character values. This, however, means that the interpretation will differ depending on collating sequence. If, for instance, one collating sequence defines as a variant of a, while another defines it as a letter following z, then the expression [–z] is valid in the first language and invalid in the second.

In the following, all examples assume the collation sequence specified for the POSIX locale, unless another collation sequence is specifically defined.

The starting range point and the ending range point must be a collating element or collating symbol. An equivalence class expression used as a starting or ending point of a range expression produces unspecified results. An equivalence class can be used portably within a bracket expression, but only outside the range. For example, the unspecified expression [[=e=]–f] should be given as [[=e=]e–f]. The ending range point must collate equal to or higher than the starting range point; otherwise, the expression will be treated as invalid. The order used is the order in which the collating elements are specified in the current collation definition. One-to-many mappings (see `locale(5)`) will not be performed. For example, assuming that the character eszet is placed in the collation sequence after `r` and `s`, but before `t`, and that it maps to the sequence `ss` for collation purposes, then the expression [r–s] matches only `r` and `s`, but the expression [s–t] matches `s`, `beta`, or `t`.

The interpretation of range expressions where the ending range point is also the starting range point of a subsequent range expression (for instance [a–m–o]) is undefined.

The hyphen character will be treated as itself if it occurs first (after an initial ^, if any) or last in the list, or as an ending range point in a range expression. As examples, the expressions [–ac] and [ac–] are equivalent and match any of the characters `a`, `c`, or –; [^ –ac] and [^ac–] are equivalent and match any characters except `a`, `c`, or –; the expression [%– –] matches any of the characters between % and – inclusive; the expression [– –@] matches any of the characters between – and @ inclusive; and the expression [a– –@] is invalid, because the letter `a` follows the symbol – in the POSIX locale. To use a hyphen as the starting range point, it must either come first in the bracket expression or be specified as a collating symbol, for example: [][.–.]–0], which matches either a right bracket or any character or collating element that collates between hyphen and 0, inclusive.

If a bracket expression must specify both – and], the] must be placed first (after the ^, if any) and the – last within the bracket expression.

Note: Latin-1 characters such as or are not printable in some locales, for example, the `ja` locale.

BREs Matching Multiple Characters

The following rules can be used to construct BREs matching multiple characters from BREs matching a single character:

1. The concatenation of BREs matches the concatenation of the strings matched by each component of the BRE.

2. A *subexpression* can be defined within a BRE by enclosing it between the character pairs \(and \) . Such a subexpression matches whatever it would have matched without the \(and \), except that anchoring within subexpressions is optional behavior; see `BRE Expression Anchoring`, below. Subexpressions can be arbitrarily nested.
3. The *back-reference* expression *n* matches the same (possibly empty) string of characters as was matched by a subexpression enclosed between \(and \) preceding the *n*. The character *n* must be a digit from 1 to 9 inclusive, *n*th subexpression (the one that begins with the *n*th \(and ends with the corresponding paired \)). The expression is invalid if less than *n* subexpressions precede the *n*. For example, the expression `^\(.*\)\1$` matches a line consisting of two adjacent appearances of the same string, and the expression `\(a\)*\1` fails to match `a`. The limit of nine back-references to subexpressions in the RE is based on the use of a single digit identifier. This does not imply that only nine subexpressions are allowed in REs. The following is a valid BRE with ten subexpressions:

   ```
   \(\(\(ab\)*c\)*d\)\(ef\)*\(gh\)\{2\}\(ij\)*\(kl\)*\(mn\)*\(op\)*\(qr\)*
   ```

4. When a BRE matching a single character, a subexpression or a back-reference is followed by the special character asterisk (*), together with that asterisk it matches what zero or more consecutive occurrences of the BRE would match. For example, `[ab]*` and `[ab][ab]` are equivalent when matching the string `ab`.
5. When a BRE matching a single character, a subexpression, or a back-reference is followed by an *interval expression* of the format \{*m*\}, \{*m*,\} or \{*m*,*n*\}, together with that interval expression it matches what repeated consecutive occurrences of the BRE would match. The values of *m* and *n* will be decimal integers in the range $0 \le m \le n \le$ {`RE_DUP_MAX`}, where *m* specifies the exact or minimum number of occurrences and *n* specifies the maximum number of occurrences. The expression \{*m*\} matches exactly *m* occurrences of the preceding BRE, \{*m*,\} matches at least *m* occurrences and \{*m*,*n*\} matches any number of occurrences between *m* and *n*, inclusive.

   For example, in the string `abababccccccd`, the BRE c\{3\} is matched by characters seven to nine, the BRE \(ab\)\{4,\} is not matched at all and the BRE c\{1,3\}d is matched by characters ten to thirteen.

The behavior of multiple adjacent duplication symbols (* and intervals) produces undefined results.

BRE Precedence

The order of precedence is as shown in the following table:

BRE Precedence (from high to low)	
collation-related bracket symbols	[= =] [: :] [. .]
escaped characters	\<*special character*>
bracket expression	[ ]
subexpressions/back-references	\(\) *n*

single-character-BRE duplication	* \{*m*,*n*\}
concatenation	
anchoring	^ $

BRE Expression Anchoring

A BRE can be limited to matching strings that begin or end a line; this is called *anchoring*. The circumflex and dollar sign special characters will be considered BRE anchors in the following contexts:

1. A circumflex (`^`) is an anchor when used as the first character of an entire BRE. The implementation may treat circumflex as an anchor when used as the first character of a subexpression. The circumflex will anchor the expression to the beginning of a string; only sequences starting at the first character of a string will be matched by the BRE. For example, the BRE `^ab` matches `ab` in the string `abcdef`, but fails to match in the string `cdefab`. A portable BRE must escape a leading circumflex in a subexpression to match a literal circumflex.
2. A dollar sign (`$`) is an anchor when used as the last character of an entire BRE. The implementation may treat a dollar sign as an anchor when used as the last character of a subexpression. The dollar sign will anchor the expression to the end of the string being matched; the dollar sign can be said to match the end-of-string following the last character.
3. A BRE anchored by both `^` and `$` matches only an entire string. For example, the BRE `^abcdef$` matches strings consisting only of `abcdef`.
4. `^` and `$` are not special in subexpressions.

Note: The Solaris implementation does not support anchoring in BRE subexpressions.

Extended Regular Expressions

The rules specifiifed for BREs apply to Extended Regular Expressions (EREs) with the following exceptions:

- The characters `|`, `+`, and `?` have special meaning, as defined below.
- The `{` and `}` characters, when used as the duplication operator, are not preceded by backslashes. The constructs `\{` and `\}` simply match the characters `{` and `}`, respectively.
- The back reference operator is not supported.
- Anchoring (`^$`) is supported in subexpressions.

EREs Matching a Single Character

An ERE ordinary character, a special character preceded by a backslash, or a period matches a single character. A bracket expression matches a single character or a single collating element. An *ERE matching a single character* enclosed in parentheses matches the same as the ERE without parentheses would have matched.

ERE Ordinary Characters

An *ordinary character* is an ERE that matches itself. An ordinary character is any character in the supported character set, except for the ERE special characters listed in `ERE Special Characters` below. The interpretation of an ordinary character preceded by a backslash (`\`) is undefined.

ERE Special Characters

An *ERE special character* has special properties in certain contexts. Outside those contexts, or when preceded by a backslash, such a character is an ERE that matches the special character itself. The extended regular expression special characters and the contexts in which they have their special meaning are:

. [\ (
: The period, left-bracket, backslash, and left-parenthesis are special except when used in a bracket expression (see RE `Bracket Expression`, above). Outside a bracket expression, a left-parenthesis immediately followed by a right-parenthesis produces undefined results.

)
: The right-parenthesis is special when matched with a preceding left-parenthesis, both outside a bracket expression.

* + ? {
: The asterisk, plus-sign, question-mark, and left-brace are special except when used in a bracket expression (see RE `Bracket Expression`, above). Any of the following uses produce undefined results:
  - if these characters appear first in an ERE, or immediately following a vertical-line, circumflex or left-parenthesis
  - if a left-brace is not part of a valid interval expression.

|
: The vertical-line is special except when used in a bracket expression (see RE `Bracket Expression`, above). A vertical-line appearing first or last in an ERE, or immediately following a vertical-line or a left-parenthesis, or immediately preceding a right-parenthesis, produces undefined results.

^
: The circumflex is special when used:
  - as an anchor (see ERE `Expression Anchoring`, below).
  - as the first character of a bracket expression (see RE `Bracket Expression`, above).

$
: The dollar sign is special when used as an anchor.

Periods in EREs

A period (`.`), when used outside a bracket expression, is an ERE that matches any character in the supported character set except NUL.

ERE Bracket Expression

The rules for ERE Bracket Expressions are the same as for Basic Regular Expressions; see RE `Bracket Expression`, above).

EREs Matching Multiple Characters

The following rules will be used to construct EREs matching multiple characters from EREs matching a single character:

1. A *concatenation of EREs* matches the concatenation of the character sequences matched by each component of the ERE. A concatenation of EREs enclosed in parentheses matches whatever the concatenation without the parentheses matches. For example, both the ERE `cd` and the ERE `(cd)` are matched by the third and fourth character of the string `abcdefabcdef`.

2. When an ERE matching a single character or an ERE enclosed in parentheses is followed by the special character plus-sign (+), together with that plus-sign it matches what one or more consecutive occurrences of the ERE would match. For example, the ERE `b+(bc)` matches the fourth to seventh characters in the string `acabbbcde; [ab]` + and `[ab][ab]*` are equivalent.
3. When an ERE matching a single character or an ERE enclosed in parentheses is followed by the special character asterisk (*), together with that asterisk it matches what zero or more consecutive occurrences of the ERE would match. For example, the ERE `b*c` matches the first character in the string `cabbbcde,` and the ERE `b*cd` matches the third to seventh characters in the string `cabbbcdebbbbbbcdbc`. And, `[ab]*` and `[ab][ab]` are equivalent when matching the string `ab`.
4. When an ERE matching a single character or an ERE enclosed in parentheses is followed by the special character question-mark (?), together with that question-mark it matches what zero or one consecutive occurrences of the ERE would match. For example, the ERE `b?c` matches the second character in the string `acabbbcde`.
5. When an ERE matching a single character or an ERE enclosed in parentheses is followed by an *interval expression* of the format {*m*}, {*m*,} or {*m*,*n*}, together with that interval expression it matches what repeated consecutive occurrences of the ERE would match. The values of *m* and *n* will be decimal integers in the range $0 \le m \le n \le$ `{RE_DUP_MAX}`, where *m* specifies the exact or minimum number of occurrences and *n* specifies the maximum number of occurrences. The expression {*m*} matches exactly *m* occurrences of the preceding ERE, {*m*,} matches at least *m* occurrences and {*m*,*n*} matches any number of occurrences between *m* and *n*, inclusive.

For example, in the string `abababccccccd` the ERE c{3} is matched by characters seven to nine and the ERE (ab){2,} is matched by characters one to six.

The behavior of multiple adjacent duplication symbols (`+, *, ?` and intervals) produces undefined results.

ERE Alternation

Two EREs separated by the special character vertical-line (|) match a string that is matched by either. For example, the ERE a((bc)|d) matches the string abc and the string ad. Single characters, or expressions matching single characters, separated by the vertical bar and enclosed in parentheses, will be treated as an ERE matching a single character.

ERE Precedence

The order of precedence will be as shown in the following table:

ERE Precedence (from high to low)	
collation-related bracket symbols	[= =] [: :] [. .]
escaped characters	\<*special character*>
bracket expression	[ ]

grouping	( )
single-character-ERE duplication	* + ? {*m*,*n*}
concatenation	
anchoring	^ $
alternation	|

For example, the ERE `abba | cde` matches either the string `abba` or the string `cde` (rather than the string `abbade` or `abbcde`, because concatenation has a higher order of precedence than alternation).

ERE Expression Anchoring

An ERE can be limited to matching strings that begin or end a line; this is called *anchoring*. The circumflex and dollar sign special characters are considered ERE anchors when used anywhere outside a bracket expression. This has the following effects:

1. A circumflex (^) outside a bracket expression anchors the expression or subexpression it begins to the beginning of a string; such an expression or subexpression can match only a sequence starting at the first character of a string. For example, the EREs ^ab and (^ab) match ab in the string abcdef, but fail to match in the string cdefab, and the ERE a^b is valid, but can never match because the `a` prevents the expression ^b from matching starting at the first character.
2. A dollar sign (`$`) outside a bracket expression anchors the expression or subexpression it ends to the end of a string; such an expression or subexpression can match only a sequence ending at the last character of a string. For example, the EREs ef$ and (ef$) match ef in the string abcdef, but fail to match in the string cdefab, and the ERE e$f is valid, but can never match because the `f` prevents the expression e$ from matching ending at the last character.

関連項目

`localedef(1)`, `regcomp(3C)`, `attributes(5)`, `environ(5)`, `locale(5)`, `regexp(5)`

名前

regexp, compile, step, advance – simple regular expression compile and match routines

形式

```
#define INIT declarations
#define GETC(void) getc code
#define PEEKC(void) peekc code
#define UNGETC(void) ungetc code
#define RETURN(ptr) return code
#define ERROR(val) error code

extern char *loc1, *loc2, *locs;

#include <regexp.h>

char *compile(char *instring, char *expbuf, const char *endfug, int eof);

int step(const char *string, const char *expbuf);

int advance(const char *string, const char *expbuf);
```

機能説明

Regular Expressions (REs) provide a mechanism to select specific strings from a set of character strings. The Simple Regular Expressions described below differ from the Internationalized Regular Expressions described on the regex(5) manual page in the following ways:

- only Basic Regular Expressions are supported
- the Internationalization features—character class, equivalence class, and multi-character collation—are not supported.

The functions `step()`, `advance()`, and `compile()` are general purpose regular expression matching routines to be used in programs that perform regular expression matching. These functions are defined by the `<regexp.h>` header.

The functions `step()` and `advance()` do pattern matching given a character string and a compiled regular expression as input.

The function `compile()` takes as input a regular expression as defined below and produces a compiled expression that can be used with `step()` or `advance()`.

Basic Regular Expressions

A regular expression specifies a set of character strings. A member of this set of strings is said to be matched by the regular expression. Some characters have special meaning when used in a regular expression; other characters stand for themselves.

The following *one-character RE*s match a *single* character:

1.1 An ordinary character (*not* one of those discussed in 1.2 below) is a one-character RE that matches itself.

1.2 A backslash (`\`) followed by any special character is a one-character RE that matches the special character itself. The special characters are:

a. `.`, `*`, `[` , and `\` (period, asterisk, left square bracket, and backslash, respectively), which are always special, *except* when they appear within square brackets (`[ ]`; see 1.4 below).

b. `^` (caret or circumflex), which is special at the *beginning* of an *entire* RE (see 4.1 and 4.3 below), or when it immediately follows the left of a pair of square brackets (`[ ]`) (see 1.4 below).

c. `$` (dollar sign), which is special at the `end` of an *entire* RE (see 4.2 below).

d. The character used to bound (that is, delimit) an entire RE, which is special for that RE (for example, see how slash (`/`) is used in the `g` command, below.)

1.3 A period (`.`) is a one-character RE that matches any character except new-line.

1.4 A non-empty string of characters enclosed in square brackets (`[ ]`) is a one-character RE that matches *any one* character in that string. If, however, the first character of the string is a circumflex (`^`), the one-character RE matches any character *except* new-line and the remaining characters in the string. The `^` has this special meaning *only* if it occurs first in the string. The minus (`–`) may be used to indicate a range of consecutive characters; for example, `[0–9]` is equivalent to `[0123456789]`. The `–` loses this special meaning if it occurs first (after an initial `^`, if any) or last in the string. The right square bracket (`]`) does not terminate such a string when it is the first character within it (after an initial `^`, if any); for example, `[ ]a–f]` matches either a right square bracket (`]`) or one of the ASCII letters `a` through `f` inclusive. The four characters listed in 1.2.a above stand for themselves within such a string of characters.

The following rules may be used to construct REs from one-character REs:

2.1 A one-character RE is a RE that matches whatever the one-character RE matches.

2.2 A one-character RE followed by an asterisk (`*`) is a RE that matches `0` or more occurrences of the one-character RE. If there is any choice, the longest leftmost string that permits a match is chosen.

2.3 A one-character RE followed by `\{`*m*`\}`, `\{`*m*`,\}`, or `\{`*m*`,`*n*`\}` is a RE that matches a *range* of occurrences of the one-character RE. The values of *m* and *n* must be non-negative integers less than 256; `\{`*m*`\}` matches *exactly m* occurrences; `\{`*m*`,\}` matches *at least m* occurrences; `\{`*m*`,`*n*`\}` matches *any number* of occurrences *between m* and *n* inclusive. Whenever a choice exists, the RE matches as many occurrences as possible.

2.4 The concatenation of REs is a RE that matches the concatenation of the strings matched by each component of the RE.

2.5 A RE enclosed between the character sequences `\ (` and `\ )` is a RE that matches whatever the unadorned RE matches.

2.6 The expression `\` *n* matches the same string of characters as was matched by an expression enclosed between `\(` and `\)` *earlier* in the same RE. Here *n* is a digit; the sub-expression specified is that beginning with the *n*-th occurrence of `\(` counting from the left. For example, the expression `^\(.*\)\1$` matches a line consisting of two repeated appearances of the same string.

An RE may be constrained to match words.

3.1 `\<` constrains a RE to match the beginning of a string or to follow a character that is not a digit, underscore, or letter. The first character matching the RE must be a digit, underscore, or letter.

3.2 `\>` constrains a RE to match the end of a string or to precede a character that is not a digit, underscore, or letter.

An *entire RE* may be constrained to match only an initial segment or final segment of a line (or both).

4.1 A circumflex (`^`) at the beginning of an entire RE constrains that RE to match an *initial* segment of a line.

4.2 A dollar sign (`$`) at the end of an entire RE constrains that RE to match a *final* segment of a line.

4.3 The construction `^`*entire RE* `$` constrains the entire RE to match the entire line.

The null RE (for example, `//`) is equivalent to the last RE encountered.

Addressing with REs

Addresses are constructed as follows:

1. The character "`.`" addresses the current line.
2. The character "`$`" addresses the last line of the buffer.
3. A decimal number *n* addresses the *n*-th line of the buffer.
4. *'x* addresses the line marked with the mark name character *x*, which must be an ASCII lower-case letter (`a–z`). Lines are marked with the `k` command described below.
5. A RE enclosed by slashes (`/`) addresses the first line found by searching *forward* from the line *following* the current line toward the end of the buffer and stopping at the first line containing a string matching the RE. If necessary, the search wraps around to the beginning of the buffer and continues up to and including the current line, so that the entire buffer is searched.
6. A RE enclosed in question marks (`?`) addresses the first line found by searching *backward* from the line *preceding* the current line toward the beginning of the buffer and stopping at the first line containing a string matching the RE. If necessary, the search wraps around to the end of the buffer and continues up to and including the current line.

7. An address followed by a plus sign (+) or a minus sign (–) followed by a decimal number specifies that address plus (respectively minus) the indicated number of lines. A shorthand for .+5 is .5.
8. If an address begins with + or –, the addition or subtraction is taken with respect to the current line; for example, –5 is understood to mean .–5.
9. If an address ends with + or –, then 1 is added to or subtracted from the address, respectively. As a consequence of this rule and of Rule 8, immediately above, the address – refers to the line preceding the current line. (To maintain compatibility with earlier versions of the editor, the character ^ in addresses is entirely equivalent to –.) Moreover, trailing + and – characters have a cumulative effect, so — refers to the current line less 2.
10. For convenience, a comma (,) stands for the address pair 1,$, while a semicolon (;) stands for the pair .,$.

Characters With Special Meaning

Characters that have special meaning except when they appear within square brackets ([ ]) or are preceded by \ are: ., *, [, \ . Other special characters, such as $ have special meaning in more restricted contexts.

The character ^ at the beginning of an expression permits a successful match only immediately after a newline, and the character $ at the end of an expression requires a trailing newline.

Two characters have special meaning only when used within square brackets. The character – denotes a range, [*c*–*c*], unless it is just after the open bracket or before the closing bracket, [–*c*] or [*c*–] in which case it has no special meaning. When used within brackets, the character ^ has the meaning *complement of* if it immediately follows the open bracket (example: [^*c*]); elsewhere between brackets (example: [*c*^]) it stands for the ordinary character ^.

The special meaning of the \ operator can be escaped only by preceding it with another \ , for example \\ .

Macros

Programs must have the following five macros declared before the #include <regexp.h> statement. These macros are used by the compile() routine. The macros GETC, PEEKC, and UNGETC operate on the regular expression given as input to compile().

GETC	This macro returns the value of the next character (byte) in the regular expression pattern. Successive calls to GETC should return successive characters of the regular expression.
PEEKC	This macro returns the next character (byte) in the regular expression. Immediately successive calls to PEEKC should return the same character, which should also be the next character returned by GETC.
UNGETC	This macro causes the argument c to be returned by the next call to GETC and PEEKC. No more than one character of pushback is ever needed and this

character is guaranteed to be the last character read by GETC. The return value of the macro UNGETC(c) is always ignored.

RETURN(*ptr*) This macro is used on normal exit of the compile() routine. The value of the argument *ptr* is a pointer to the character after the last character of the compiled regular expression. This is useful to programs which have memory allocation to manage.

ERROR(*val*) This macro is the abnormal return from the compile() routine. The argument *val* is an error number (see ERRORS below for meanings). This call should never return.

compile() The syntax of the compile() routine is as follows:

compile(*instring*, *expbuf*, *endbuf*, *eof*)

The first parameter, *instring*, is never used explicitly by the compile() routine but is useful for programs that pass down different pointers to input characters. It is sometimes used in the INIT declaration (see below). Programs which call functions to input characters or have characters in an external array can pass down a value of (char *)0 for this parameter.

The next parameter, *expbuf*, is a character pointer. It points to the place where the compiled regular expression will be placed.

The parameter *endbuf* is one more than the highest address where the compiled regular expression may be placed. If the compiled expression cannot fit in (endbuf–expbuf) bytes, a call to ERROR(50) is made.

The parameter *eof* is the character which marks the end of the regular expression. This character is usually a /.

Each program that includes the <regexp.h> header file must have a #define statement for INIT. It is used for dependent declarations and initializations. Most often it is used to set a register variable to point to the beginning of the regular expression so that this register variable can be used in the declarations for GETC, PEEKC, and UNGETC. Otherwise it can be used to declare external variables that might be used by GETC, PEEKC and UNGETC. (See EXAMPLES below.)

step(), advance() The first parameter to the step() and advance() functions is a pointer to a string of characters to be checked for a match. This string should be null terminated.

The second parameter, *expbuf*, is the compiled regular expression which was obtained by a call to the function compile().

The function step() returns non-zero if some substring of *string* matches the regular expression in *expbuf* and 0 if there is no match. If there is a match, two external character pointers are set as a side effect to the call to step(). The variable loc1 points to the first

character that matched the regular expression; the variable `loc2` points to the character after the last character that matches the regular expression. Thus if the regular expression matches the entire input string, `loc1` will point to the first character of *string* and `loc2` will point to the null at the end of *string*.

The function `advance()` returns non-zero if the initial substring of *string* matches the regular expression in *expbuf*. If there is a match, an external character pointer, `loc2`, is set as a side effect. The variable `loc2` points to the next character in *string* after the last character that matched.

When `advance()` encounters a * or `\{ \}` sequence in the regular expression, it will advance its pointer to the string to be matched as far as possible and will recursively call itself trying to match the rest of the string to the rest of the regular expression. As long as there is no match, `advance()` will back up along the string until it finds a match or reaches the point in the string that initially matched the * or `\{ \}`. It is sometimes desirable to stop this backing up before the initial point in the string is reached. If the external character pointer `locs` is equal to the point in the string at sometime during the backing up process, `advance()` will break out of the loop that backs up and will return zero.

The external variables `circf`, `sed`, and `nbra` are reserved.

使用例

例 1 Using Regular Expression Macros and Calls

The following is an example of how the regular expression macros and calls might be defined by an application program:

```
#define INIT       register char *sp = instring;
#define GETC()     (*sp++)
#define PEEKC()    (*sp)
#define UNGETC(c)  (——sp)
#define RETURN(c)  return;
#define ERROR(c)   regerr()

#include <regexp.h>
 . . .
      (void) compile(*argv, expbuf, &expbuf[ESIZE],'\0');
 . . .
      if (step(linebuf, expbuf))
                        succeed;
```

診断

The function `compile()` uses the macro `RETURN` on success and the macro `ERROR` on failure (see above). The functions `step()` and `advance()` return non-zero on a successful match and zero if there is no match. Errors are:

11 range endpoint too large.

16 bad number.

25 \ *digit* out of range.

36 illegal or missing delimiter.

41 no remembered search string.

42 \(\) imbalance.

43 too many \(.

44 more than 2 numbers given in \{ \}.

45 } expected after \.

46 first number exceeds second in \{ \}.

49 [ ] imbalance.

50 regular expression overflow.

関連項目 regex(5)

名前

resource_controls – プロジェクトデータベースから使用可能なリソース制御

機能説明

リソース制御の機能は、プロジェクトデータベースから構成されます。project(4) を参照してください。リソース制御は、次のユーティリティーから設定および変更できます:

- prctl(1)
- projadd(1M)
- projmod(1M)
- rctladm(1M)

プログラムでは、setrctl(2) を使用してリソース制御値を設定します。

前のリソース制御に加えて、pooladm(1M) および poolcfg(1M) ユーティリティーからアクセス可能なリソースプールがあります。プログラムでは、リソースプールは libpool(3LIB) ライブラリから操作できます。

使用可能なリソース制御を次に示します:

process.max-address-space
: このプロセスで使用できる、セグメントサイズについて集計されたアドレス空間の最大量で、バイト数で表現されます。

process.max-core-size
: このプロセスによって作成されるコアファイルの最大サイズで、バイト数で表現されます。

process.max-cpu-time
: このプロセスで使用できる最大 CPU 時間で、秒数で表現されます。

process.max-data-size
: このプロセスで使用できる最大ヒープメモリーで、バイト数で表現されます。

process.max-file-descriptor
: このプロセスで使用できる最大のファイル記述子インデックスで、整数で表現されます。

process.max-file-size
: このプロセスでの書き込みに使用できる最大ファイルオフセット。バイト数で表現されます。

processs.max-itimers
: インターバルタイマーの許容可能な最大数で、整数で表現されます。

process.max-msg-messages
: メッセージキュー上のメッセージの最大数 (msgget() の時点でリソース制御からコピーされる値) で、整数で表現されます。

`process.max-msg-qbytes`
メッセージキュー上のメッセージの最大バイト数 (`msgget()` の時点でリソース制御からコピーされる値) で、バイト数で表現されます。

`process.max-port-events`
イベントポートごとのイベントの許容可能な最大数で、整数で表現されます。

`process.max-sem-nsems`
セマフォーセットごとに許可されるセマフォーの最大数で、整数で表現されます。

`process.max-sem-ops`
`semop` 呼び出しごとに許可されるセマフォー操作の最大数 (`semget()` の時点でリソース制御からコピーされる値)。整数で表現され、操作の数を指定します。

`process.max-sigqueue-size`
プロセスが保有できるキューに入れられた未処理のシグナルの最大数。

`process.max-stack-size`
このプロセスで使用できる最大スタックメモリーセグメントで、バイト数で表現されます。

`project.cpu-caps`
プロジェクトで使用できる CPU リソースの最大容量。使用される単位は、プロジェクト内のすべてのユーザースレッドで使用できる 1 つの CPU の割合 (%) です。整数で表現されます。リアルタイムスケジューリングクラスで実行されているスレッドに上限は適用されません。このリソース制御は、`syslog` アクションをサポートしていません。

`project.cpu-shares`
公平配分スケジューラで使用するためにプロジェクトに付与される CPU 共有の数 (FSS(7) を参照)。使用される単位は共有の数 (整数) です。このリソース制御は、`syslog` アクションをサポートしていません。

`project.max-contracts`
プロジェクト内で許可される契約の最大数で、整数で表現されます。

`project.max-crypto-memory`
暗号化操作に使用できるカーネルメモリーの最大容量。カーネル内のバッファーおよびセッション関連の構造体の割り当ては、このリソース制御に対してチャージされます。

`project.max-locked-memory`
デバイスドライバおよびユーザープロセス (D/ISM を含む) によってロックされた物理メモリーの合計容量で、バイト数で表現されます。

`project.max-lwps`
プロジェクトで同時に使用できる LWP の最大数で、整数で表現されます。

`project.max-msg-ids`
プロジェクトに対して許可されるメッセージキュー ID の最大数で、整数で表現されます。

`project.max-port-ids`
イベントポートの許容可能な最大数で、整数で表現されます。

`project.max-processes`
プロジェクトで同時に使用できるプロセスの最大数で、整数で表現されます。

`project.max-sem-ids`
プロジェクトに対して許可されるセマフォー ID の最大数で、整数で表現されます。

`project.max-shm-ids`
プロジェクトに対して許可される共有メモリー ID の最大数で、整数で表現されます。

`project.max-shm-memory`
プロジェクトに対して許可される共有メモリーの合計容量で、バイト数で表現されます。

`project.max-tasks`
プロジェクト内で許容可能なタスクの最大数で、整数で表現されます。

`project.pool`
指定されたリソースプールをプロジェクトにバインドします。

`rcap.max-rss`
プロジェクト内のプロセスで使用できる物理メモリーの合計容量 (バイト単位)。

`task.max-cpu-time`
このタスクのプロセスで使用できる最大 CPU 時間で、秒数で表現されます。

`task.max-lwps`
このタスクのプロセスで同時に使用できる LWP の最大数で、整数で表現されます。

`task.max-processes`
タスクで同時に使用できるプロセスの最大数で、整数で表現されます。

次のゾーン規模のリソース制御を使用できます:

`zone.cpu-cap`
ゾーンで使用できる CPU 時間に対して制限を設定します。使用される単位は、ゾーン内のすべてのユーザースレッドで使用できる 1 つの CPU の割合 (%) です。整数で表現されます。上限が設けられたゾーン内のプロジェクトに独自の上限がある場合は、最小値が優先されます。このリソース制御は、`syslog` アクションをサポートしていません。

`zone.cpu-shares`
ゾーンに対する公平配分スケジューラ (FSS) CPU 共有の数に制限を設定します。CPU 配分は、まずゾーンに対して割り当てられたあとで、`project.cpu-shares` エントリの指定に従って、ゾーン内のプロジェクトに分配されます。整数で表現されます。このリソース制御は、`syslog` アクションをサポートしていません。

`zone.max-locked-memory`
ゾーンで使用できるロックされた物理メモリーの合計量。

`zone.max-lofi`
ゾーンで使用できる `lofi(7D)` デバイスの最大数。

`zone.max-lwps`
1 つのゾーン内の LWP が多すぎて、ほかのゾーンに影響を与えることがないようにすることで、リソースの分離を強化します。ゾーンの LWP の合計数は、`project.max-lwps` エントリを使用すると、ゾーン内のプロジェクト間でさらに再分割できます。整数で表現されます。

`zone.max-msg-ids`
ゾーンに対して許可されるメッセージキュー ID の最大数で、整数で表現されます。

`zone.max-processes`
ゾーンで同時に使用できるプロセスの最大数で、整数で表現されます。

`zone.max-sem-ids`
ゾーンに対して許可されるセマフォー ID の最大数で、整数で表現されます。

`zone.max-shm-ids`
ゾーンに対して許可される共有メモリー ID の最大数で、整数で表現されます。

`zone.max-shm-memory`
ゾーンに対して許可される共有メモリーの合計容量で、バイト数で表現されます。

`zone.max-swap`
このゾーンのユーザープロセスのアドレス空間マッピングと `tmpfs` マウントで消費できるスワップの合計量。

`zones(5)` を参照してください。

リソース制御で使用される単位

リソース制御は、サイズ (バイト)、時間 (秒)、または数 (整数) の単位で表現できます。これらの単位は、次に指定されている文字列を使用します。

Category	Res Ctrl Type String	Modifier	Scale
Size	bytes	B	1
		KB	2^10

		MB	2^20
		GB	2^30
		TB	2^40
		PB	2^50
		EB	2^60
Time	seconds	s	1
		Ks	10^3
		Ms	10^6
		Gs	10^9
		Ts	10^12
		Ps	10^15
		Es	10^18
Count	integer	none	1
		K	10^3
		M	10^6
		G	10^9
		T	10^12
		P	10^15
		Es	10^18

リソース制御に倍率値を使用できます。次の例は、倍率付きのしきい値を示します。

```
task.max-lwps=(priv,1K,deny)
```

`project` ファイルでは、`1K` の値が `1000` に展開されます:

```
task.max-lwps=(priv,1000,deny)
```

2 番目の例では、より大きな倍率値が使用されています:

```
process.max-file-size=(priv,5G,deny)
```

`project` ファイルでは、`5G` の値が `5368709120` に展開されます:

```
process.max-file-size=(priv,5368709120,deny)
```

前の例では、上の表で指定されている倍率が使用されています。

単位修飾子 (たとえば、`5G`) が `prctl(1)`、`projadd(1M)`、および `projmod(1M)` コマンドで受け入れられていることに注意してください。プロジェクトデータベース自体で単位修飾子を使用することはできません。

リソース制御値と特権レベル

リソース制御に対するしきい値によって、ローカルアクションをトリガーできるか、または (ロギングなどの) グローバルアクションを実行できる時点が構成されます。

リソース制御に対する各しきい値は、特権レベルに関連付けられている必要があります。特権レベルは、次の 3 つの種類のいずれかである必要があります:

basic
呼び出し元プロセスの所有者が変更できます。

privileged
現在のプロセス (sys_resource 特権が必要) または prctl(1) (proc_owner 特権が必要) が変更できます。

system
オペレーティングシステムインスタンスの期間中は固定されます。

リソース制御は、システムまたはリソースプロバイダで定義された 1 つの system 値を持つことが保証されます。system 値は、オペレーティングシステムの現在の実装で提供できるリソースの量を表します。

特権値はいくつでも定義できます。基本値は 1 つだけ許可されます。特権値を指定しないで実行される操作には、デフォルトで、基本レベルの特権が割り当てられます。

リソース制御値の特権レベルは、リソース制御ブロックの特権フィールドで RCTL_BASIC、RCTL_PRIVILEGED、または RCTL_SYSTEM として定義されます。詳細は、setrctl(2) のマニュアルページを参照してください。prctl コマンドを使用すると、基本レベルおよび特権レベルに対応付けられている値を変更できます。

privileged の特権レベルを指定する場合は、priv の省略形を使用できます。例:

```
task.max-lwps=(priv,1K,deny)
```

リソース制御値に対応付けられた大域アクションと局所アクション

リソース制御値に関するアクションには、グローバルとローカルの 2 つのカテゴリがあります。

グローバルアクションは、システム上のすべてのリソース制御のリソース制御値に適用されます。rctladm(1M) を使用して次のアクションを実行できます:

- アクティブなシステムリソース制御のグローバルな状態を表示します。
- 大域ログ作成アクションを設定します。

リソース制御に対応付けられた大域ログ作成アクションは、無効にしたり有効にしたりできます。syslog アクションの程度を設定するには、重要度を syslog=*level* のように割り当てます。*level* に設定できる値は次のとおりです。

- debug
- info
- notice
- warning
- err
- crit
- alert
- emerg

デフォルトでは、リソース制御の違反は大域ログ作成では記録されません。

ローカルアクションは、制御値を超えようとしているプロセスに対して実行されます。リソース制御に設定された各しきい値に対して、1 つ以上のアクションを対応付けることができます。局所アクションには、3 つの種類があります。none、deny、および signal= です。これら 3 つのアクションは、次のように使用されます。

none
しきい値を超える量のリソース要求に対して、何のアクションも行いません。このアクションは、アプリケーションの進行に影響を与えることなく、リソースの使用状況を監視するのに役立ちます。また、リソース制御を超えたにもかかわらず、同時に、しきい値を超えたプロセスが影響を受けていないときに表示されるグローバルメッセージを有効にすることもできます。

deny
しきい値を超える量のリソース要求を拒否できます。たとえば、task.max-lwps リソース制御に拒否アクションが指定されている場合、新しいプロセスが制御値を超えると fork() システムコールは失敗します。fork(2) を参照してください。

signal=
リソース制御値を超えたときに大域シグナルメッセージを送信するアクションを有効にすることができます。プロセスがしきい値を超えると、プロセスにシグナルが送信されます。プロセスがさらにリソースを消費しても、追加のシグナルが送信されることはありません。使用可能なシグナルを次に一覧表示します。

すべてのリソース制御にすべてのアクションを適用できるわけではありません。たとえば、プロセスは、その所属先のプロジェクトに割り当てられている CPU 配分を超えることはできません。したがって、project.cpu-shares リソース制御に deny アクションを適用することはできません。

実装上の制限により、しきい値に設定できるアクションは、各制御の大域プロパティーによって制限されます。(rctladm(1M) を参照。)使用可能なシグナルアクションの一覧を次のリストに示します。シグナルについての詳細は、signal(3HEAD) を参照してください。

リソース制御値に使用可能なシグナルを次に示します:

SIGABRT
プロセスを終了します。

SIGHUP
ハングアップシグナルを送信します。開いた回線上でキャリアが検出されなくなったときに発生します。シグナルは、端末を制御しているプロセスグループに送信されます。

SIGTERM
プロセスを終了します。ソフトウェアによって送信される終了シグナルです。

SIGKILL
プロセスを終了し、プログラムを強制終了します。

SIGSTOP
プロセスを停止します。ジョブ制御シグナル。

SIGXRES
リソース制御の制限超過です。リソース制御機能によって生成されます。

SIGXFSZ
プロセスを終了します。ファイルサイズの制限超過です。RCTL_GLOBAL_FILE_SIZE プロパティー (process.max-file-size) を持つリソース制御でのみ使用できます。rctlblk_set_value(3C) を参照してください。

SIGXCPU
プロセスを終了します。CPU 時間の制限超過です。RCTL_GLOBAL_CPUTIME プロパティー (process.max-cpu-time) を持つリソース制御でのみ使用できます。rctlblk_set_value(3C) を参照してください。

リソース制御のフラグとプロパティー

システム上のリソース制御には、それぞれ特定のプロパティーセットが対応付けられています。このプロパティーセットは、一連のフラグとして定義されます。これらのフラグは、そのリソースが制御されているすべてのインスタンスに対応付けられます。大域フラグは変更できませんが、rctladm(1M) または setrctl(2) システムコールを使用するとフラグを取得できます。

ローカルフラグは、特定のプロセスまたはプロセス集合に対するリソース制御の特定のしきい値について、デフォルトの動作と構成を定義します。あるしきい値のローカルフラグが、同じリソース制御で定義されている別のしきい値の動作に影響することはありません。ただし、大域フラグは、特定の制御に対応付けられているすべての値の動作に影響します。ローカルフラグは、prctl コマンドまたは setrctl システムコールを使用すると、対応する各大域フラグで指定された制約内で変更できます。setrctl(2) を参照してください。

ローカルフラグ、大域フラグ、およびそれらの定義の詳細な一覧については、rctlblk_set_value(3C) を参照してください。

特定のリソース制御がしきい値に達したときのシステムの動作を確認するには、rctladm を使ってそのリソース制御の大域フラグを表示します。たとえば、process.max-cpu-time の値を表示するには、次を入力します:

```
$ rctladm process.max-cpu-time
process.max-cpu-time  syslog=off [ lowerable no-deny cpu-time inf seconds ]
```

大域フラグは次を示します:

lowerable
この制御の特権値を下げるのに、スーパーユーザー特権を必要としません。

no-deny
しきい値を超えても、リソースへのアクセスは拒否されません。

cpu-time
このリソースのしきい値に達したら、SIGXCPU を送信できます。

seconds
リソース制御の時間。

リソース制御のローカル値とアクションを表示するには、prctl コマンドを使用します。例:

```
$ prctl -n process.max-cpu-time $$
    process 353939: -ksh
    NAME    PRIVILEGE    VALUE    FLAG   ACTION              RECIPIENT
 process.max-cpu-time
        privileged   18.4Es    inf   signal=XCPU                 -
        system       18.4Es    inf   none
```

両方のしきい値に対して max (RCTL_LOCAL_MAXIMAL) フラグが設定され、このリソース制御に対して inf (RCTL_GLOBAL_INFINITE) フラグが定義されています。inf の値は無限大です。この値は制限を与えません。したがって、構成されているように、両方のしきい値は無限大値を意味し、これらの値を上回ることはありません。

リソース制御の適用

1 つのリソースには、複数のリソース制御を設定できます。リソース制御は、プロセスモデルの包含レベルごとに 1 つずつ設定できます。同じリソース上の異なるコンテナレベルでリソース制御がアクティブな場合、まず、もっとも小さいコンテナの制御が実行されます。このため、両方の制御が同時に検出された場合は、task.max-cpu-time の前に process.max-cpu-time に対するアクションが実行されます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
インタフェースの安定性	確実

関連項目

prctl(1)、pooladm(1M)、poolcfg(1M)、projadd(1M)、projmod(1M)、rctladm(1M)、setrctl(2)、r

『Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management』

名前

sgml, solbook – Standard Generalized Markup Language

機能説明

Standard Generalized Markup Language (SGML) is the ISO standard 8879:1986 that describes a syntax for marking up documents with tags that describe the purpose of the text rather than the appearance on the page. This form of markup facilitates document interchange between different platforms and applications. SGML allows the management of information as data objects rather than text on a page.

In an SGML document the main structural components are called `elements`. The organization and structure of a document and the meaning of elements are described in the Document Type Definition (DTD). Elements are the *tags* that identify the content. Element names may be descriptive of the content for ease of use. For example `<para>` for paragraphs. Elements can have `attributes` which are used to modify or refine the properties or characteristics of the element. Within the DTD a valid context for each element is defined and a framework is provided for the types of elements that constitute a compliant document.

Another component of the DTD is `entities`. Entities are a collection of characters that can be referenced as a unit. Entities are similar to constants in a programming language such as C. They can be defined and referenced. An entity can represent one character or symbol which does not appear on a standard keyboard, a word or group of words, or an entire separate sgml marked-up file. Entities allow reuse of standard text.

There is no single standard DTD , but the de facto standard for the computer industry is the DocBook DTD , developed and maintained by the Davenport Group. Within Sun, the SolBook DTD , which is a proper subset of DocBook DTD , is used when writing reference manual pages. The SolBook DTD contains a number of tags that are designed for the unique needs of the reference pages.

SolBook Elements

Elements are defined with a hierarchical structure that gives a structure to the document. The following is a description of some of the elements from the SolBook DTD which are used for reference pages.

DOCTYPE

The first line in an SGML file that identifies the location of the DTD that is used to define the document. The `<!DOCTYPE` string is what the SGML -aware `man(1)` command uses to identify that a file is formatted in SGML rather than `nroff(1)`.

RefEntry

The top layer element that contains a reference page is `<refentry>`. All of the text and other tags must be contained within this tag.

RefMeta

The next tag in a reference page is `<refmeta>`, which is a container for several other tags. They are:

<refentrytitle> This is the title of the reference page. It is equivalent to the name of the reference page's file name, without the section number extension.

<manvolnum> This is the section number that the reference page resides in. The contents may be a text entity reference.

<refmiscinfo> There are one or more <refmiscinfo> tags which contain *meta* information. Meta information is information about the reference page. The <refmiscinfo> tag has the class attribute. There are four classes that are routinely used.

- date — This is the date that the file was last modified. By consensus this date is changed only when the technical information on the page changes and not simply for an editorial change.
- sectdesc — This is the section title of the reference page; for example User Commands. The value of this attribute may be a text entity reference.
- software — This is the name of the software product that the topic discussed on the reference page belongs to. For example UNIX commands are part of the SunOS x.x release. The value of this attribute may be a text entity reference.
- arch — This is the architectural platform limitation of the subject discussed on the reference page. If there are no limitations the value used is generic. Other values are sparc and x86.
- copyright — This attribute contains the Sun Microsystems copyright. Any other copyrights that may pertain to the individual reference page file should be entered as separate <refmiscinfo> entries. The value of this attribute may be a text entity reference.

RefNameDiv This tag contains the equivalent information to the .TH macro line in an nroff(1) reference page. <refnamediv> contains three tags. These tags contain the text that is before and after the '–' (dash) on the NAME line.

<refname> These are the names of the topics that are discussed in the file. There may be more than one <refname> for a page. The first <refname> must match the name of the file and the <refentrytitle>. If there are more than one <refname> tags, each is separated by a ',' (comma). The comma is generated by the publisher of sgml files, so it should not be typed. This is referred to as *auto-generated* text.

<refpurpose> The text after the dash on the NAME line is contained in this tag. This is a short summary of what the object or objects described on the reference page do or are used for. The dash is also auto-generated and should not be typed in.

<refdiscriptor> In some cases the <refentrytitle> is a general topic descriptor of a group of related objects that are discussed on the same page. In this case the first tag after the <refnamediv> is a <refdiscriptor>. The <refname> tags follow. Only one <refdiscriptor> is allowed, and it

should match the <refentrytitle>.

RefSynopsisDiv

The SYNOPSIS line of the reference page is contained by this tag. There is a <title> that usually contains an entity reference. The text is the word SYNOPSIS. There are several tags within <refsynopsisdiv> that are designed specifically for the type of synopsis that is used in the different reference page sections. The three types are:

<cmdsynopsis>	Used for commands and utilities pages.
<funcsynopsis>	Used for programming interface pages.
<synopsis>	Used for pages that do not fall into the other two categories.

RefSect1

This tag is equivalent to the .SH nroff macro. It contains a <title> element that is the title of the reference page section. Section names are the standard names such as DESCRIPTION, OPTIONS, PARAMETERS, SEE ALSO, and others. The contents of the <title> may be a text entity reference.

RefSect2

This tag is equivalent to the .SS nroff macro. It contains a <title> element that contains the text of the sub-section heading. <refsect2> tags may also be used within a <refsynopsisdiv> as a sub-section heading for the SYNOPSIS section.

Block Elements

There are a number of block elements that are used for grouping text. This is a list of some of these elements.

<para>	This tag is used to contain a paragraph of text.
<variablelist>	This tag is used to create two column lists. For example descriptions for command options, where the first column lists the option and the second column describes the option.
<orderedlist>	An list of items in a specific order.
<itemizedlist>	A list of items that are marked with a character such as a bullet or a dash.
<literallayout>	Formatted program output as produced by a program or command. This tag is a container for lines set off from the main text in which line breaks, tabs, and leading white space are significant.
<programlisting>	A segment of program code. Line breaks and leading white space are significant.
<table>	This tag contains the layout and content for tabular formatting of information. <table> has a required <title>.
<informaltable>	This tag is the same as the <table> tag except the <title> is not required.

<example>	This tag contains examples of source code or usage of commands. It contains a required `<title>`.
<informalexample>	This tag is the same as the `<example>` tag except the `<title>` is not required.

Inline Elements

The inline elements are used for tagging text.

<command>	An executable program or the entry a user makes to execute a command.
<function>	A subroutine in a program or external library.
<literal>	Contains any literal string.
<parameter>	An argument passed to a computer program by a function or routine.
<inlineequation>	An untitled mathematical equation occurring in-line.
<link>	A hypertext link to text within a book, in the case of the reference manual it is used to cross reference to another reference page.
<olink>	A hypertext link used to create cross references to books other than the reference manual.
<xref>	A cross reference to another part of the same reference page.

関連項目

`man(1)`, `nroff(1)`, `man(5)`

名前

smf – サービス管理機能

機能説明

Solaris サービス管理機能は、「サービス」と呼ばれる永続的に実行されるアプリケーションを提供するためのプログラミングモデルを定義します。また、この機能は、サービスを実行するためのインフラストラクチャーも提供します。サービスは、実行中のアプリケーション、デバイスのソフトウェア状態、その他の一連のサービスのいずれかを表現できます。このフレームワーク内では、サービスは「サービスインスタンス」オブジェクトとして表現されます。これは、サービスオブジェクトの子になります。インスタンスオブジェクトは、親であるサービスオブジェクトの構成を継承またはオーバーライドできます。これにより、複数のサービスインスタンス間で構成情報を共有することができます。すべてのサービスオブジェクトとインスタンスオブジェクトは、一連の構成情報を表現した単一の「スコープ」内に格納されます。ローカル Solaris インスタンスの構成は「localhost」スコープと呼ばれますが、これが現在サポートされている唯一のスコープとなります。

各サービスインスタンスの名前は、障害管理リソース識別子 (Fault Management Resource Identifier、FMRI) に基づいて、スキーム「`svc:`」を使って付けられます。たとえば、システム起動時に起動される `syslogd(1M)` デーモンは、次のような名前を持つデフォルトサービスインスタンスです。

```
svc://localhost/system/system-log:default
svc:/system/system-log:default
system/system-log:default
```

多くのコマンドでは FMRI の省略形も使用できます。そのような例については、`svcs(1)` のマニュアルページを参照してください。

上記の例では、`default` がインスタンスの名前、`system/system-log` がサービス名です。サービス名は、スラッシュ (`/`) で区切られた複数のコンポーネントから構成される場合があります。最後のものを除くすべてのコンポーネントで、そのサービスの「カテゴリ」が構成されます。サイト固有のサービスに名前を付ける場合は、`site` で始まるカテゴリを使用するようにしてください。

サービスインスタンスは、有効化または無効化されます。すべてのサービスは、`svcadm(1M)` コマンドを使って有効化または無効化できます。

システム上の管理対象サービスインスタンスを一覧表示するには、`svcs(1)` コマンドを使用します。

管理者が標準の場所にあるマニフェストまたはプロファイルに基づくエンティティーを削除すると、そのエンティティーはマスクされて、SMF への通常のクエリーで表示されなくなります。マスクされたエンティティーを見つけるには `svccfg listcust` を使用し、削除するには `svccfg` の `delcust` サブコマンドを使用します。詳細は、`svccfg(1M)` を参照してください。

依存関係

サービスインスタンスは、サービス、インスタンス、ファイルなどの一連の entities (エンティティー) に対する依存関係を持つ可能性があります。依存関係により、サービスがいつ起動され、いつ自動的に停止されるかが左右されます。サービスが有効化されていてもその依存関係が満たされていない場合、そのサービスはオフライン状態に保たれます。その依存関係が満たされると、そのサービスは起動されます。起動が成功すると、そのサービスはオンライン状態に移行します。サービスやインスタンスとは異なり、ファイルの依存関係は、ファイルが作成または削除されるたびに動的に評価されるということはありません。これらは 1 回だけ評価されます。

依存関係が満たされるかどうかは、サービスの grouping (グループ化) によって決まります。

require_all　引用されているすべてのサービスが実行中 (オンライン、機能低下のいずれか) の場合、または指定されているすべてのファイルが存在している場合に満たされます。

require_any　引用されているサービスのいずれかが実行中 (オンライン、機能低下のいずれか) の場合、または指定されているファイルの少なくとも 1 つが存在している場合に満たされます。

optional_all　引用されているサービスが実行中 (オンライン、機能低下のいずれか) の場合、または管理作業が行われないためにそれらのサービスが実行されていない場合 (つまり、管理作業が行われないために起動されない依存関係に対して待機状態にあるために、それらのサービスが無効、保守、存在しない、またはオフライン状態になっている場合) に満たされます。不完全なサービスも、オプションの依存関係を満たします。

exclude_all　引用されているすべてのサービスが無効になっているか保守状態にある場合、または引用されているサービスまたはファイルが存在していない場合に満たされます。

require_all、require_any、optional_all のいずれかの依存関係から引用されている特定のサービスが、いったん実行中 (オンライン、機能低下のいずれか) になったあとで停止またはリフレッシュされた場合、SMF は、そのサービスが停止した理由とその依存関係の restart_on 属性に基づいて、サービスを停止するかどうかを決定します。

```
                  |  restart_on value
event             |  none  error restart refresh
------------------+-----------------------------
stop due to error |  no    yes   yes     yes
non-error stop    |  no    no    yes     yes
refresh           |  no    no    no      yes
```

あるサービスがエラーによって停止したとみなされるのは、コアダンプなどのハードウェアエラーやソフトウェアエラーがそのサービスで発生した場合です。exclude_all 依存関係の場合、引用されているサービスが起動され、かつ restart_on 属性が none 以外になっている場合にサービスが停止されます。

特定のサービスの依存関係を一覧表示するには svcs(1) または svccfg(1M) を使用し、変更するには svccfg(1M) を使用します。

リスタータ

各サービスは特定のリスタータによって管理されます。マスターリスタータ svc.startd(1M) は、一連のサービスインスタンスとその依存関係の状態をすべて管理します。マスターリスタータは、自身のサービスに代って各種処理を行うほか、特定アプリケーションクラス向けの特定実行環境を提供できる委任リスタータを制御します。たとえば、inetd(1M) は委任リスタータであり、入力ファイル記述子と出力ファイル記述子で表されるネットワーク接続から成る初期環境を、自身のサービスインスタンスに対して提供します。inetd(1M) に委任された各インスタンスは、オンライン状態になっています。ある特定のインスタンスのデーモンが実行されていなくても、そのインスタンスを実行することは可能です。

依存関係が満たされるには、各インスタンスがオンライン状態に移行する必要があります。このため、svc.startd(1M) は、ほかのインスタンスの起動メソッドを呼び出すか、委任リスタータにそうするように指示します。これらの処理はオーバーラップする可能性があります。

現在のサービス群およびそれらに関連付けられたリスタータを確認するには、svcs(1) を使用します。すべてのリスタータが使用する共通の構成については、smf_restarter(5) を参照してください。

メソッド

各サービスまたはサービスインスタンスは、サービスの起動、停止、およびリフレッシュ (省略可能) を行う一連のメソッドを定義する必要があります。svc.startd(1M) および類似の fork(2)-exec(2) リスタータ用のメソッド規約の詳細については、smf_method(5) を参照してください。

レガシー構成情報を取得してリポジトリ内に格納するメソッドなど、各種の管理メソッドについては、svccfg(1M) のマニュアルページを参照してください。

特定のサービスのメソッドを一覧表示したり変更したりするには、svccfg(1M) コマンドを使用します。

状態

各サービスインスタンスは常に明確に定義された特定の状態にありますが、どの状態になるかは、その依存関係、メソッドの実行結果、および契約イベントの可能性によって決まります。定義されている状態は、次のとおりです。

未初期化
: これは、すべてのサービスインスタンスの初期状態です。インスタンスは、svc.startd(1M) または適切なリスタータによって評価された結果、保守、オフライン、無効のいずれかの状態に移行します。

OFFLINE	インスタンスは有効になっていますが、まだ実行中でも実行可能でもありません。リスタータがあるサービスの起動メソッドまたはそれと同等のメソッドを正常に実行できた場合、そのインスタンスはオンライン状態に移行します。失敗した場合は通常、機能低下、保守のいずれかの状態に移行することがあります。管理作業を行うと未初期化状態に移行する可能性があります。
ONLINE	インスタンスは有効になっており、実行中であるか実行可能になっています。オンライン状態の具体的な内容はアプリケーションモデルに固有であり、サービスインスタンスを管理するリスタータによって定義されます。オンラインは、適切に構成されたサービスのすべての依存関係が満たされた場合に予想される動作状態です。インスタンスで障害が発生すると、機能低下、保守のいずれかの状態に移行する可能性があります。インスタンスが依存するサービスで障害が発生すると、オフライン、機能低下のいずれかの状態に移行する可能性があります。
DEGRADED	インスタンスは有効になっており、実行中であるか実行可能になっています。ただし、通常の動作と比較すると、インスタンスはある制限された機能レベルで動作しています。インスタンスで障害が発生すると、保守状態に移行する可能性があります。インスタンスが依存するサービスで障害が発生すると、オフライン、機能低下のいずれかの状態に移行する可能性があります。機能が回復すると、オンライン状態に移行します。
保守	インスタンスは有効になっていますが、実行可能ではありません。インスタンスを保守状態から移行させるには、`svcadm clear` による管理作業が必要です。何らかの管理操作を実施中の場合、この保守状態に一時的に達する場合があります。
無効	インスタンスは無効になっています。サービスを有効化するとオフライン状態に移行し、最終的には、すべての依存関係が満たされた時点でオンライン状態に移行します。
レガシー実行	この状態は、サービス管理機能によって管理されていないレガシーインスタンスを表します。この状態のインスタンスはある時点で起動されたものですが、それが実行中かどうかはわかりません。この機能を使って行えるのは、インスタンスの監視だけであり、ほかの状態に移行させることはできません。

状態の移行には、結果的に元の状態に戻るようなものもあります。

イベント通知

SMF では、SNMP または SMTP を使用することで状態の遷移を通知できます。状態遷移に関する情報イベントが発行され、`snmp-notify(1M)` や `smtp-notify(1M)` などの通知デーモンによって処理されます。無効になっているサービスの SMF 状態遷移では、通知は生成されません。ただし、遷移の最終状態が「無効」で、その遷移

に関する通知パラメータが存在する場合に限り、通知が生成されます。遷移の初期状態と最終状態が同じである場合、通知は生成されません。

通知パラメータ

SMF 状態遷移によって生成される情報イベントを除き、FMA イベントの通知パラメータは `svc:/system/fm/notify-params:default` に保存されます。これらは、サービスまたは遷移中のサービスのインスタンスに保存されます。SMF 状態遷移によって生成されるイベントの通知パラメータは、システム全体のパラメータとして `svc:/system/svc/global:default` で設定できます。システム全体の通知パラメータは、`scf_instance_get_pg_composed(3SCF)` と同様の合成検索が遷移中のインスタンスに見つからない場合に使用されます。通知パラメータは `svccfg(1M)` を使用すると操作できます。DTD に記述されている `notification_parameters` 要素を使用すると、サービスマニフェストまたはサービスプロファイルで通知パラメータを構成できます。次に例を示します。

```
<notification_parameters>
     <event  value='from-online' />
     <type name='smtp' active="false">
        <parameter name='to'>
            <value_node value='root@local' />
            <value_node value='admin-alias@eng' />
        </parameter>
     </type>
     <type name='snmp' />
</notification_parameters>
```

events は、SMF 状態遷移セットをコンマで区切ったリストか、または FMA イベントクラスをコンマで区切ったリストです。*events* に SMF 状態遷移セットと FMA イベントクラスを混在させることはできません。

FMA サブシステムで診断される問題のライフサイクルを、初期の診断から暫定的な更新、最後の問題解決まで、`problem-{diagnosed,updated,repaired,resolved}` の各タグを利用して記述できます。これらのタグは、基になる FMA プロトコルイベントクラス (すべて `list.*` 階層に含まれる) の別名ですが、後者は通知の構成に使用しないようにしてください。

`problem-diagnosed`
: 新しい問題が FMA サブシステムによって診断されました。診断には、1 つ以上の疑わしいリソースからなるリストが含まれています。これらのリソースは、それ以上のエラーの発生を防ぐために、自動的に隔離されている可能性があります (適切な場合)。問題はイベントペイロードの UUID で識別されます。この問題の解決ライフサイクルを表す追加のイベントには、一致する UUID が使用されます。

`problem-updated`
: 問題の診断に含まれていた疑わしいリソースの 1 つ以上が、修復または交換されたか、または疑いがなくなりました (あるいは、再度障害が発生しました)。ただし、障害の発生したリソースが少なくとも 1 つリストに残っています。修復

は、fmadm コマンド行 (fmadm repaired、fmadm acquit、fmadm replaced) の結果として、あるいは部品シリアル番号の変更の検出などを通して自動的に検出された可能性があります。

problem-repaired
: 問題の診断に含まれていた疑わしいリソースのすべてが、修復または解決されたか、または疑いがなくなりました。この段階では、一部またはすべてのリソースがまだ隔離されている可能性があります。

problem-resolved
: 問題の診断に含まれていた疑わしいリソースのすべてが、修復または解決されたか、疑いがなくなりました。さらに、隔離も解除されました (オフラインになっていた疑わしい CPU がオンラインに戻った場合など。通常、この隔離解除の処理は自動的に行われます)。

状態遷移セットは次のように定義されます。

to-<state>	遷移の最終状態が <state> である、すべての遷移のセット。
from-<state>	遷移の初期状態が <state> である、すべての遷移のセット。
<state>	遷移の初期状態が <state> である、すべての遷移のセット。
all	すべての遷移のセット。

state の有効な値は、maintenance、offline、disabled、online、および degraded です。遷移セットの定義の例は、maintenance、 from-online、to-degraded などです。

プロパティーとプロパティーグループ

これまでに説明した依存関係、メソッド、委任リスタータ、およびインスタンス状態は、サービスまたはサービスインスタンスのプロパティーまたはプロパティーグループとして表現されます。サービスまたはサービスインスタンスは、任意の数のプロパティーグループを、アプリケーションデータの格納先として持つことができます。プロパティーグループをこのような方法で使用すれば、リポジトリがこの機能内のすべてのデータに対して提供する属性を、アプリケーションの構成情報から導き出すことができます。また、アプリケーションは、service_bundle(4) DTD の適切なサブセットを使ってフレームワーク内の自身の構成データを表現することもできます。

プロパティーの検索は合成されます。あるプロパティーグループとプロパティーの組み合わせがサービスインスタンス上で見つからなかった場合、libscf(3LIB) の大部分のコマンドや高レベルインタフェースは、その同じプロパティーとプロパティーグループの組み合わせを、そのインスタントを含むサービス上で検索します。これにより、共通の構成をサービスインスタンス間で共有することが可能になります。この合成は、サービスインスタンスとその親であるサービスとの間の一種の継承関係として捉えることができます。

プロパティーは、承認されていないプロセスによる変更から保護されます。smf_security(5) を参照してください。

general プロパティーグループ

general プロパティーグループはすべてのサービスインスタンスに適用されます。次のプロパティーが含まれています。

enabled (boolean)
: インスタンスが有効になっているかどうかを指定します。このプロパティーがインスタンス上に存在しない場合、SMF は、そのインスタンスのリスタータにインスタンスの存在について通知しません。

restarter (fmri)
: このサービスのリスタータ。詳細については、「リスタータ」のセクションを参照してください。このプロパティーが設定されていない場合は、システムのデフォルトのリスタータが使用されます。

complete (astring)
: このサービスが完了したか、または開始すべきでない部分的な定義であるかを示します。このプロパティーは、マニフェストのインポート時に自動的に設定されます。あるいは、このプロパティーを持たないインスタンスがテンプレート定義に対して正常に検証された場合は (scf_tmpl_validate_fmri(3SCF) を参照)、このインスタンスが有効になるときに svcadm(1M) によってこのプロパティーが作成されます。

レイヤー

リポジトリは、管理カスタマイズ、現在の状態、および標準の場所にあるファイルからのデフォルト値の組み合わせで構成されています。SMF 管理のファイルシステムの位置にあるマニフェストによって定義されているサービス、インスタンス、プロパティーグループ、およびプロパティーは、リポジトリ内で常に正確に表現されます。管理者またはほかのプログラムによって実行時に行われたカスタマイズは、捕獲されてリポジトリに保管されます。

プロパティーはリポジトリ内で、マニフェスト、プロファイル、および管理カスタマイズからの異なる設定を反映した異なる値を持つ場合があります。デフォルトでユーザーおよびサービスに提供される値は、 レイヤーと呼ばれる単純な優先順位スキームによって調停されます。

SMF によって 4 つのレイヤーが追跡されます。優先順位の高い順に示すと、次のようになります。

admin
: SMF コマンドまたはライブラリの対話型使用によって実行されたすべての変更。このレイヤーは優先順位がもっとも高くなります。

site-profile
: /etc/svc/profile/site ディレクトリのファイルか、レガシーの /etc/svc/profile/site.xml および /var/svc/profile/site.xml ファイルのすべての値。

system-profile　システムプロファイルの場所 /etc/svc/profile/generic.xml および /etc/svc/profile/platform.xml のすべての値

manifest　システムマニフェストの場所 /lib/svc/manifest または /var/svc/manifest のすべての値。

個々のレイヤー内におけるプロパティーの競合は許可されません。admin レイヤーに競合するプロパティーがある場合、以前のプロパティーが単純に上書きされます。同じプロパティーが複数のファイルによってほかのレイヤーで配信され、高いレイヤーで設定されない場合、インスタンス全体に競合状態のタグが付けられ、競合中の定義が削除されるか高いレイヤーでプロパティーが設定されるまでは、svc.startd(1M) によって開始されません。svccfg および svcprop などの単一の値を要求するその他の libscf 利用者は、すべての適切な値からランダムなプロパティー設定を取得します。競合中のどの値が返されるかは保証されません。

スナップショット　リポジトリ内の各インスタンスに関する履歴データが、サービス管理機能によって管理されます。このデータは、管理上の検査やロールバック向けの読み取り専用スナップショットとして利用可能となります。利用可能なスナップショットタイプは次のとおりです。

initial　管理者によって作成されたかパッケージインストール中に生成されたインスタンスの初期構成。

previous　元に戻す管理操作を実行する際に取得された、その時点における構成。

running　インスタンスの実行中の構成。

start　オンライン状態への正常移行中に取得された構成。

svccfg(1M) コマンドを使用すれば、スナップショットを操作できます。

特殊なプロパティーグループ　プロパティーグループの中には、「非永続的」とマークされているものがあります。それらのグループはスナップショット内にバックアップされず、その内容はシステムブート中にクリアされます。そのようなグループは一般に、システムの再起動時に消えてもかまわないようなアクティブプログラム状態を保持します。

構成リポジトリ　サービスやサービスインスタンスに関連付けられたプロパティーに加え、各サービスインスタンスの現在の状態が、svc.configd(1M) が管理するシステムリポジトリ内に格納されます。

サービス管理機能データ用のリポジトリを管理するには、svc.configd(1M) を使用します。

サービスバンドル、マニフェスト、およびプロファイル　構成リポジトリ内に格納されている、サービスまたはサービスインスタンスに関連付けられた情報は、XML ベースのファイルとしてエクスポートできます。サービ

スバンドルと呼ばれるそれらの XML ファイルは移植性に優れており、バックアップ用途に適しています。サービスバンドルは次のいずれかのタイプに分類されます。

`manifests` 特定のサービス群またはサービスインスタンス群に関連付けられたプロパティーをすべて含んだファイル。

`profiles` 一連のサービスインスタンスと各インスタンスの enabled プロパティー (general プロパティーグループの `boolean` 型プロパティー) の値を含んだファイル。

プロファイルには、サービスおよびインスタンスのプロパティーの構成値も含まれることがあります。プロファイルにテンプレート要素を定義することはできません。

プロファイルでは、`service_bundle(4)` で説明されている DTD の要素の緩和されたセットを使用できます。これらを使用するには、`DOCTYPE` エントリに次の定義を追加するようにしてください:

```
<!ENTITY % profile "INCLUDE">
<!ENTITY % manifest "IGNORE">
```

特定のリポジトリに対してサービスバンドルのインポート、エクスポートを行うには、`svccfg(1M)` コマンドを使用します。サービスバンドルのファイル形式や作成時のガイドラインについては、`service_bundle(4)` を参照してください。

マイルストーン `smf` マイルストーンは、複数のサービス依存関係を集約するサービスです。通常、マイルストーンは、それ自体で有用な機能を果たすことはありませんが、ほかのサービスが利用できるようにシステム対応状況の特定の状態を宣言します。たとえば、`name-services` マイルストーンは、単に現在有効になっているネームサービスに依存します。

レガシー起動スクリプト `/etc/rc?.d` ディレクトリ内の起動プログラムは、対応する実行レベルのマイルストーンの一部として実行されます。

```
/etc/rcS.d    milestone/single-user:default

/etc/rc2.d    milestone/multi-user:default

/etc/rc3.d    milestone/multi-user-server:default
```

各プログラムの実行は特定の機能限定版のサービスインスタンスとして表現され、プログラムのパスに基づいて命名されます。これらのインスタンスは、特殊な状態であるレガシー実行状態に保たれます。

これらのインスタンスは enabled プロパティー (general プロパティーグループの `boolean` 型プロパティー) を持たず、一般に `svcadm(1M)` コマンドを使って操作することもできません。これらのプログラムについては、エラー診断や再起動は行われません。

関連項目 svcs(1),
inetd(1M)、snmp-notify(1M)、smtp-notify(1M)、svcadm(1M)、svccfg(1M)、svc.configd(1M)、s
libscf(3LIB)、scf_tmpl_validate_fmri(3SCF)、strftime(3C)、contract(4)、service_bundle(4)、

名前

smf_bootstrap – サービス管理機能のブート、パッケージ化、および互換性の動作

機能説明

サービス管理機能は、サービスマニフェストの配信、サービスマニフェストの変更の組み込み、サービス構成の安定性の記述、サービス構成の優先指定の使用、およびサービスプロファイルの使用に関する規約を定めます。

ブート時のマニフェストの読み込み

システムのブート時と、管理者またはプログラムが実行するときは常に、標準のディレクトリツリー /lib/svc/manifest および /var/svc/manifest からのマニフェストが処理されます。

```
$ svcadm restart manifest-import
```

以前インポートされなかったか、最後にインポートされてから変更されたマニフェストが処理されます。マニフェストが変更されたかどうかを判断するためにハッシュが使用されます。

標準の場所にあるマニフェストがはじめてインポートされると、マニフェストのプロパティー、インスタンス、およびサービスが、manifest レイヤーの一部としてリポジトリに追加されます。

標準の場所にあるマニフェストは、更新されると自動的にインポートされます。新しいサービスおよびインスタンスは追加され、変更されたプロパティーはアップグレードされ、削除されたサービス、インスタンス、およびプロパティーは削除されます。

マニフェストはブート中に 2 つの段階で処理されます。

サービス svc:/system/early-manifest-import:default は、最初のマニフェスト処理を担当する擬似サービスです。このサービスは /lib/svc/manifest ディレクトリツリーにあるマニフェストだけを処理します。この処理は、svc.startd(1M) がサービスを初期化する前に行われるため、/lib/svc/manifest で配信されているサービスは常に最新の定義で起動されます。これは擬似サービスなので svcadm(1M) コマンドは無視されます。ただし、svcs(1) を使用して、ステータスの監視とログファイル情報の取得を行うことはできます。

svc:/system/manifest-import:default サービスは 2 番目のマニフェスト処理を担当し、まず /lib/svc/manifest、次に /var/svc/manifest の順に、これら両方のディレクトリツリーからマニフェストファイルをインポートします。

/var/svc/manifest のサポートは、system/early-manifest-import:default の導入より前にそのディレクトリツリーで配信されていたマニフェストに対する互換性サポートです。/var/svc/manifest で配信されているサービスには、アップグレード関連の問題が発生する可能性があります。つまり、更新されたマニフェストがまだインポートされていないために、サービスが古いリポジトリ構成で起動する可能性があります。同様に、マニフェストファイルがまだ処理されていないために、新たに追加されたサービスが使用できなくなったり、削除されたサービスがブート中に

起動したりすることもあります。このような問題を避けるために、開発者にはマニフェストを /lib/svc/manifest に移動することを強くお勧めします。

このセクションで説明している操作では、*.xml ファイルのみが考慮されます。

プロファイルの適用

プロファイルは early-manifest-import サービスおよび manifest-import サービスによっても適用されます。

/etc/svc/profile/generic.xml および /etc/svc/profile/platform.xml にあるシステム配信プロファイルは system-profile レイヤーにインポートされます。

/etc/svc/profile/site ディレクトリにあるサイト固有プロファイルと、レガシーサイトファイル /etc/svc/profile/site.xml および /var/svc/profile/site.xml は、site-profile レイヤーにインポートされます。

管理者は次を実行することによって、これらのプロファイルが再適用されることを要求できます。

```
$ svcadm restart manifest-import
```

プロファイルによって定義されるプロパティー、インスタンス、およびサービスの動作は、マニフェストによって定義される動作と同じです。

このセクションで説明している操作では、*.xml ファイルのみが考慮されます。

パッケージ化操作中のマニフェストの処理

パッケージ内のサービスマニフェストは、クラス manifest によって識別されます。サービスマニフェストのインストールと削除を行うクラスアクションスクリプトは、パッケージ化サブシステムに含まれています。pkg install が呼び出されると、サービスマニフェストがインポートされます。

pkg uninstall が呼び出されると、無効になっているマニフェスト内のインスタンスが削除されます。マニフェスト内のインスタンスで、オンライン状態または機能低下状態になっているものは、まず無効化され、そのあとで削除されます。また、インスタンスが残っていないマニフェスト内のサービスもすべて削除されます。

安定性の宣言

マニフェストで配信する各サービスグループおよび各プロパティーグループには、attributes(5) の定義に基づく安定性レベルを宣言するようにしてください。アプリケーション開発者は、サービスやオブジェクトの存在またはそのコンポーネントを基に機能を開発する場合に、安定性レベルがわかっていれば、リリースが変わっても動作するかどうかを判断しやすくなります。

smf(5) のコンテキストでは、安定性の値は、そのサービスのリリースが変わるときに想定される、プロパティーグループ内のプロパティーに対する変更の範囲も示します。これにはそのサービスのパッチも含まれます。次の 2 つのセクションでは、これについて詳細に説明します。

プロパティーグループの削除

service_bundle(4) ドキュメント型定義には、サービスマニフェスト内の各プロパティーグループに適用できる delete 属性があります。 true に設定されている場合、delete 属性は svccfg(1M) およびほかのマニフェストインポートツールに対し、このプロパティーグループをリポジトリから削除するように指示します。delete 属性が存在しないか false に設定されている場合、そのプロパティーグループはリポジトリに保持されます。

「安定」または「開発中」と宣言されているプロパティーグループは削除されません。「不安定」と宣言されているプロパティーグループは、リリースが変わるときに削除される可能性があります。

関連項目

svcs(1), svcadm(1M), svccfg(1M), svc.startd(1M), libscf(3LIB), service_bundle(4), attributes(5), smf(5), smf_security(5)

pkg(1)

注意事項

smf(5) の現在のバージョンでは、複数のリポジトリはサポートされていません。

名前

smf_method – サービス管理フレームワークのメソッド規約

機能説明

サービス管理フレームワーク smf(5) で svc.startd(1M) によって管理されるサービスのクラスは、単純な fork(2)-exec(2) モデルに適合するアプリケーションから成ります。svc.startd(1M) マスターデーモンとほかのリスタータは fork(2)-exec(2) モデルをサポートしており、追加の機能を備えている場合もあります。svc.startd(1M) デーモンとほかのリスタータにとっては、サービスインスタンスのアクティブ化、操作、または検査を行うメソッドがこのマニュアルページで説明されている規約に従っている必要があります。

呼び出しの形式

メソッド呼び出しの形式は規約で規定されていません。メソッドの呼び出しは、デーモンまたはサービスを提供するほかの実行可能バイナリの直接呼び出しの場合もあります。実行可能スクリプトなどの間接実行可能ファイルを使用する場合は、規約では次の形式が推奨されています。

```
/path/to/method_executable abbr_method_name
```

この推奨形式の *abbr_method_name* は、start や stop など、サポートされているメソッドです。リスタータでサポートされているメソッドセットについては、そのリスタータの関連ページで説明されています。svc.startd(1M) デーモンは start、stop、および refresh メソッドをサポートしています。

リスタータには、このページで説明するメソッドのほかにも各種のメソッドが定義されていることがあります。そのような拡張も含めた規約はリスタータによって定義され、ここで説明する規約とは異なる場合があります。

環境変数

リスタータは、メソッド呼び出しのコンテキストを決定する 4 つの環境変数をメソッドに提供します。

SMF_FMRI
: メソッド呼び出しの対象であるインスタンスのサービス障害管理リソース識別子 (FMRI)。

SMF_METHOD
: 呼び出されるメソッドの完全な名前。start や stop など。

SMF_RESTARTER
: メソッドを呼び出すリスタータのサービス FMRI。

SMF_ZONENAME
: メソッドが実行されているゾーンの名前。これは zonename(1) コマンドでも取得できます。

メソッドで永続的プロセスを呼び出す場合は、その前にこれらの変数を環境から削除するようにしてください。サービスの作成者が Bourne 互換のシェルスクリプトを使用して後述のインクルードファイルにサービスメソッドを合成できるように、簡易シェル関数 smf_clear_env が用意されています。

後述のとおり、メソッドコンテキストによってほかの環境変数が設定されることもあります。

メソッドの定義

メソッドは、`method` タイプのプロパティーグループの 3 つのプロパティーで最小限定義されます。

これらのプロパティーは次のとおりです。

exec (*astring*)
: メソッドの実行可能ファイルの文字列。

timeout_seconds (*count*)
: メソッドがタイムアウトするまでの秒数。詳細については、「タイムアウト」のセクションを参照してください。

type (*astring*)
: メソッドのタイプ。現在は常に `method` に設定されます。

メソッドコンテキストを定義して、メソッドの実行環境をより詳細に指定することもできます。詳細については、「メソッドコンテキスト」のセクションを参照してください。

メソッドトークン

リスタータ `svc.startd` によってメソッドの `exec` 文字列に一連のトークンが定義されている場合は、それらが解析され、適切な値で展開されます。ほかのリスタータはメソッドトークンをサポートしていない場合があります。inet サービスの委任リスタータ `inetd(1M)` は、次のメソッド展開をサポートしていません。

`%%`
: %

`%r`
: リスタータの名前。`svc.startd` など。

`%m`
: 呼び出されるメソッドの完全な名前。`start` や `stop` など。

`%s`
: サービスの名前

`%i`
: インスタンスの名前

`%f`
: インスタンスの FMRI

`%{prop[:,]}`
: プロパティーの値。`prop` は、プロパティーの FMRI、プロパティーグループ名とプロパティー名を `/` で区切ったもの、または `application` プロパティーグループ内のプロパティー名です。これらの値のあとに , (コンマ) または : (コロン) が続く場合があります。ある場合、その区切り文字が複数の値を区切るために使用さ

れます。ない場合は、スペースが使用されます。次に示すシェルのメタキャラクタが文字列値の中に見つかった場合は、\(バックスラッシュ) でクォートされます。

```
; & ( ) | ^ < > newline space tab  \  " '
```

展開が正しくないとメソッドの失敗を招きます。

メソッドコマンドの代わりに、2 つの明示的なトークンを使用できます。

`:kill [-signal]`
: 指定されたシグナルをプライマリインスタンス契約のすべてのプロセスに送信します。デフォルトのシグナルは SIGTERM です。常に SMF_EXIT_OK を返します。このトークンは、一般的な pkill の呼び出しの代わりに使用するようにしてください。

`:true`
: 常に SMF_EXIT_OK を返します。このトークンは、リスタータには必要であっても、特定のサービス実装には必要のないメソッドに使用するようにしてください。

終了と終了ステータス

start メソッドは、サービスインスタンスが要求に応答できる状態になるか、あるいは機能するまで、終了を遅らせる必要があります。

次の終了ステータスコードが <libscf.h> とシェルサポートファイルに定義されています。

SMF_EXIT_OK	0	メソッドは処理を正常に実行して終了しました。
SMF_EXIT_ERR_FATAL	95	メソッドで致命的な失敗が発生し、管理者の操作なしでは回復不可能です。
SMF_EXIT_ERR_CONFIG	96	回復不可能な構成エラーです。一般に、この終了ステータスが返されるのは、有効になっているサービスインスタンスに必要な構成ファイルが見つからない場合です。
SMF_EXIT_ERR_NOSMF	99	メソッドは誤って smf(5) 機能の外部で呼び出されました。smf(5) の機能に依存しているサービスは、このステータス値で終了するようにしてください。
SMF_EXIT_ERR_PERM	100	メソッドには、ファイルのアクセス権、特権、承認、資格情報といった形式の特権が必要ですが、呼び出し時には使用できませんでした。

SMF_EXIT_ERR_OTHER	0 以外	メソッドから 0 以外の終了ステータスが返された場合、それは不明なエラーとして扱われます。一連の不明なエラーは、リスタータで、あるいはリスタータに代わって、障害として診断できます。

上で説明した終了コードに加えて、メソッドでは次の終了コードを、smf_method_exit(3SCF) および smf_include.sh から使用可能な smf_method_exit() と組み合わせて使用できます:

SMF_EXIT_TEMP_DISABLE	101	メソッドは正常に終了し、一時的な無効をリクエストします。
SMF_EXIT_TEMP_TRANSIENT	105	メソッドは正常に終了し、そのサービスモデルが「一時的」であるかのように処理されることをリクエストします。

正確な終了コードを使用することで、担当のリスタータでエラー応答を診断して、再起動が適切と考えられる断続的なエラーと、管理者の操作を必要とする永続的なエラーとに分類できます。

タイムアウト

メソッドごとに個別のタイムアウトを秒単位で指定できます。メソッドのタイムアウトは timeout_seconds プロパティーで指定されます。

タイムアウトは、メソッドがハングアップしているか、または進行していないことをサービスのリスタータが判断するための最後の手段として使用されます。タイムアウトが経過すると、多くのリスタータはそのサービスを保守状態にします。svc.startd(1M) を参照してください。メソッドは進行しているが、メモリー、CPU、または I/O 負荷によってシステムの応答が一時的に非常に遅くなったときの早期エラーを避けるため、タイムアウトを指定する場合は誤差の範囲を大きくすることをお勧めします。

1 - 2 秒しかかからないと予想されるメソッドの場合、適切な開始値は 60 秒です。30 秒かかることが多いメソッドの場合は 300 秒 (5 分) が適切です。日常的に長い時間のかかるメソッドの場合は、必要に応じて値を大きくしてください。

管理者の介入を促すために早期エラーが必要な場合は、短いタイムアウトを使用できます。管理者の介入がサービスをクリアーしてメソッドを再度開始させるだけの可能性が高い場合は、長いタイムアウトを考慮します。

timeout_seconds が 0 に設定されている場合、そのサービスにはタイムアウトがありません。この設定は推奨されませんが、どうしてもそれを必要とするサービスには使用できます。-1 もタイムアウトがないことを指定するために受け入れられますが、非推奨です。

シェルプログラムのサポート

前述の終了ステータス値を定義する一連の環境変数が、/lib/svc/share/smf_include.sh ファイル内の簡易シェル関数で提供されます。このファイルは Bourne シェルスクリプトであり、任意の Bourne 互換シェルのソース演算子を介したインクルードに適しています。

SMF メソッドとして機能するスクリプトおよび /etc/init.d スクリプトの合成を容易にするために、smf_present() シェル関数が用意されています。smf(5) 機能を使用できない場合、smf_present() は 0 以外の終了ステータスを返します。

そのようなスクリプトの構造の例を次に示します。

```
if smf_present; then
      # Shell code to run application as managed service
      ....

      smf_clear_env
else
      # Shell code to run application as /etc/init.d script
      ....
fi
```

この例では、用意されている両方の簡易関数の使用方法が示されています。

メソッドコンテキスト

サービス管理機能は、fork(2)-exec(2) モデルサービスが実行されるコンテキストを設定するための共通のメカニズムを提供します。

必要なメソッドコンテキストはサービス開発者が指定するようにしてください。セキュリティーが損なわれることをできるだけ防ぐために、すべてのサービスインスタンスは最低限の特権で実行されるべきです。

メソッドコンテキストには次のプロパティーを含めることができます。

use_profile
: user、group、privileges、および limit_privileges プロパティーの代わりにプロファイルを使用するかどうかを指定するブール値。

environment
: メソッドの環境に挿入する環境変数を、いくつかの NAME=value 文字列で指定します。

profile
: RBAC (役割に基づくアクセス制御) プロファイルの名前。メソッドの実行可能ファイルとの組み合わせで、exec_attr(4) 内のエントリを識別します。

user
: 数値またはテキスト形式のユーザー ID。

group
: 数値またはテキスト形式のグループ ID。

`supp_groups`
補助グループメンバーシップを数値またはテキスト形式の ID で指定する、省略可能な文字列。

`privileges`
privileges(5) で定義されている特権セットを指定する、省略可能な文字列。拡張ポリシーはここで指定できます。

`limit_privileges`
privileges(5) で定義されている制限特権セットを指定する、省略可能な文字列。

`working_directory`
メソッドを起動するホームディレクトリ。トークンとして `:home` を使用すると、メソッドの起動に使用された `uid` に該当するユーザーのホームディレクトリを指定できます。このプロパティーが設定されていない場合は、`:home` が使用されます。

`corefile_pattern`
coreadm(1M) に従ってサービスに使用するコアファイルパターンを指定する、省略可能な文字列。ほとんどのリスタータにはデフォルトが用意されています。このプロパティーの設定は、グローバルなコアパターンをローカルでカスタマイズした設定よりも優先されます。

`project`
数値またはテキスト形式のプロジェクト ID。トークンとして `:default` を使用すると、メソッドの起動に使用された `uid` に該当するユーザーの、getdefaultproj(3PROJECT) で識別されるプロジェクトを指定できます。

`resource_pool`
メソッドを起動するリソースプールの名前。トークンとして `:default` を使用すると、上記の `project` 属性で示された project(4) エントリに指定されているプールを指定できます。

サービスまたはインスタンスの `method_context` プロパティーグループを指定することにより、サービスインスタンス全体のメソッドコンテキストを設定できます。メソッドプロパティーグループのメソッドコンテキストプロパティーを指定することで、メソッドがインスタンスのメソッドコンテキストをオーバーライドする場合もあります。

メソッドコンテキストの設定が正しくないと、メソッドは常に失敗しますが、環境変数が無効な場合は警告が発行されます。

上記で定義されるコンテキストに加え、fork(2)-exec(2) モデルのリスタータは、実行可能ファイルをメソッドとして呼び出すときに次の規約も使用します。

引数配列
argv[] 内の引数は、exec 文字列の /bin/sh -c の結果と一致するように設定されます。

ファイル記述子
ファイル記述子 0 は /dev/null です。ファイル記述子 1 と 2 は、サービスごとのログファイルに推奨されます。

ファイル

/lib/svc/share/smf_include.sh
終了ステータス値の定義。

/usr/include/libscf.h
終了ステータスコードの定義。

使用例

例 1 サービスログでサービス固有の構成エラーメッセージを報告します。

start メソッドでは、サービスログでサービス固有の構成エラーメッセージを報告するために smf_method_exit() を使用することもできます。

```
if [ ! -s "$my_config_file" ]; then
   smf_method_exit $SMF_EXIT_ERR_CONFIG \
       missing_or_empty_config_file \
       "$my_config_file is missing or empty"
fi
```

例 2 非大域ゾーンで起動されたときに大域ゾーンでのみ実行されるべきサービスを無効にします。

大域ゾーンでのみ実行されるべきサービスは、非大域ゾーンで起動されたときに、自身を無効にすることもできます。

```
if smf_is_nonglobalzone; then
    smf_method_exit $SMF_EXIT_TEMP_DISABLE global_zone_only \
    "$SMF_FMRI is not supported in a local zone" SUNW_OST_OSCMD
fi
```

関連項目

zonename(1)、coreadm(1M)、inetd(1M)、svccfg(1M)、svc.startd(1M)、exec(2)、fork(2)、getdef
smf_bootstrap(5)、zones(5)

注意事項

smf(5) の現在のバージョンでは、複数のリポジトリはサポートされていません。

サービスが limit_privileges とは異なる特権で root ユーザーとして起動するように構成されている場合、結果として得られるプロセスは特権に対応したものになります。これは、seteuid(<non-zero UID>) によって特権が basic 以下に下がると考えている開発者には予期しないことです。

名前

smf_restarter – サービス管理機能のリスタータ規約

機能説明

サービス管理機能のすべてのサービスインスタンスは、リスタータによって管理される必要があります。このマニュアルページでは、フレームワークのすべてのリスタータに共通する構成、機能、およびレポートの特徴について説明します。特定のリスタータに固有の特徴については、そのリスタータのマニュアルページを参照してください。

各管理対象サービスについて、リスタータはサービスインスタンスのプロパティーを取得することによって構成を判定します。リスタータは一連のプロパティーグループを利用して、サービスの現在の配置を svcs(1) などの表示ツールに伝達します。

サービスの構成

すべてのサービスに共通するリスタータの構成は general プロパティーグループにまとめられています。このグループには、次に示す必須プロパティーおよび省略可能プロパティーの設定が含まれています。

enabled
: これは必須プロパティーです。これが設定されている場合、インスタンスのリスタータはサービスを使用可能な状態に維持しようとします。

restarter
: これは省略可能プロパティーで、サービスインスタンスを管理する代替リスタータを指定できます。restarter プロパティーが空または省略されている場合、リスタータはデフォルトとして svc.startd(1M) になります。

single_instance
: これは省略可能プロパティーです。これが設定されている場合、オンラインステータスまたは機能低下ステータスに移行できるサービスインスタンスは一度に 1 つだけです。

サービスの報告機能

すべてのリスタータが restarter プロパティーグループを使用してステータスを報告します。このグループには次のプロパティーが含まれています。

next_state
: これらのプロパティーに保存されているインスタンスの現在の状態および次の状態 (移行中の場合)。取り得る状態については、smf(5) を参照してください。

auxiliary_state
: リスタータの state プロパティーとの組み合わせで完全なリスタータ固有の状態を表す明確な語句を含んだ astring。スペースは含まれません。この補助状態は常に設定できるわけではなく、どの状態からでも移行時に必ずクリアされます。リスタータごとに、使用する補助状態の正確なリストを定義する必要があります。

state_timestamp
: 現在の状態に到達した時間。

contract
: サービスインスタンスを実行しているプライマリプロセス契約 ID (存在する場合)。

関連項目 svcs(1), svc.startd(1M), service_bundle(4), smf(5), smf_method(5)

名前

smf_security – サービス管理機能のセキュリティー動作

機能説明

サービス管理機能の構成サブシステム smf(5) には、サービスの構成を変更するための特権が必要です。ユーザーに特権を付与するには、user_attr(4) および prof_attr(4) を通して、次に説明する承認をユーザーに関連付けます。rbac(5) を参照してください。

サービスおよびサービスインスタンスの操作には次の承認が使用されます。

`solaris.smf.modify`
: サービス、サービスインスタンス、またはそれらのプロパティーの追加、削除、変更、および保護されたプロパティー値の読み取りが承認されます。

プロパティーグループ承認

smf(5) 構成サブシステムは、各サービスおよびサービスインスタンスにプロパティーを関連付けます。関連するプロパティーはグループ化されています。グループは、実行メソッド、資格情報、アプリケーションデータ、リスタータの状態を表す場合があります。プロパティーグループを作成または変更する機能がある場合、オペレーティングシステム特権を必要とする可能性のあるアクションを smf(5) コンポーネントで実行させることができます。それに従って、フレームワークでは、プロパティーグループを操作するための適切な承認が要求されます。

各プロパティーグループには、その目的に対応するタイプがあります。コアとなるプロパティーグループタイプは、`method`、`dependency`、`application`、および `framework` です。追加のプロパティーグループタイプを導入することもできますが、smf(5) の拡張命名規約に適合する必要があります。ただし、次の基本承認は、コアとなるプロパティーグループタイプだけに適用されます。

`solaris.smf.modify.method`
: `method` タイプのプロパティーグループの値の変更または作成、削除、変更が承認されます。

`solaris.smf.modify.dependency`
: `dependency` タイプのプロパティーグループの値の変更または作成、削除、変更が承認されます。

`solaris.smf.modify.application`
: application タイプのプロパティーグループの値の変更、保護された値の読み取り、および作成、削除、変更が承認されます。

`solaris.smf.modify.framework`
: `framework` タイプのプロパティーグループの値の変更または作成、削除、変更が承認されます。

`solaris.smf.modify`
: サービス、サービスインスタンス、またはそれらのプロパティーの追加、削除、変更、および保護されたプロパティー値の読み取りが承認されます。

プロパティーグループ固有の承認は、そのプロパティーグループに含まれているプロパティーによって指定できます。

`modify_authorization`
承認は、プロパティーグループ内のプロパティーの追加、削除、または変更を許可します。また、プロパティーグループが保護されている場合は、そのグループからプロパティー値を取得することも許可します。

`value_authorization`
承認は、`modify_authorization` を除くプロパティーグループの任意のプロパティーに対する値の変更を許可します。また、プロパティーグループが保護されている場合は、そのグループから modify_authorization 以外の任意のプロパティー値を取得することも許可します。

`read_authorization`
承認は、プロパティーグループ内のプロパティー値の取得を許可します。プロパティーグループ内にこの名前の文字列値プロパティーが存在する場合、そのプロパティーグループは保護されています。このプロパティーは、application タイプ以外のプロパティーグループには影響しません。「保護されたプロパティーグループ」を参照してください。

前述の承認プロパティーは、そのタイプが `astring` の場合だけ使用されます。インスタンスプロパティーグループにいずれかのプロパティーがない場合でも、そのプロパティーと同じ名前を持つプロパティーグループがインスタンスのサービスにあるときは、その値が使用されます。

保護されたプロパティーグループ

通常、リポジトリ内のすべてのプロパティー値は、明示的な承認なしで任意のユーザーが読み取ることができます。framework タイプ以外のプロパティーグループは、値の保護を必要とするプロパティーを格納するために使用できます。適切な承認がない場合、これらは公開できません。プロパティーグループのステータスが保護されたステータスであることは、文字列値の `read_authorization` プロパティーの存在によって示されます。このプロパティーが存在する場合、そのプロパティーグループに含まれているプロパティーの値は、「プロパティーグループ承認」で説明されている方法によってのみ取得可能です。

機密性が高いとみなされるデータのバックアップをポリシーによって禁止している管理ドメインでは、SMF リポジトリデータベースをバックアップから除外するようにしてください。そのようなポリシーが存在しても、保護されていないプロパティー値は `svccfg(1M)` アーカイブコマンドでバックアップできるため、保護されたプロパティー値を除いたリポジトリのアーカイブを作成できます。

サービスアクション承認

サービスインスタンスに対する特定のアクションは、結果としてサービスを中断または非アクティブ化する場合があります。どのようなサービス拒否も意図的な管理操作であることを保証するために、これらのアクションには承認が必要です。このようなアクションには、refresh メソッドや restart メソッドを実行する要求、サービスインスタンスを保守に配置するなどの非稼働状態にする要求があります。次の承認は、このようなアクションの要求を許可します。

`solaris.smf.manage`　任意のサービスインスタンスの再起動、リフレッシュ、またはその他の状態変更を要求することが承認されます。

さらに、`general/action_authorization` プロパティーで追加の承認を指定して、そのサービスインスタンスに対するサービスアクションの要求を許可できます。このプロパティーを変更するには、`solaris.smf.manage` 承認が必要です。

定義済み権利プロファイル

smf(5) の一般的な処理を操作するための承認をグループ化した 2 つの権利プロファイルが用意されています。

サービス管理
: サービスマネージャーはリポジトリ内のすべてのサービスを任意の方法で操作できます。これは、`solaris.smf.manage` 承認と `solaris.smf.modify` 承認に対応します。

  pkg(1) コマンドを使用して、サービスマニフェスト内にサービスインベントリを含んでいるソフトウェアパッケージの追加や削除を行うには、少なくともサービス管理プロファイルが必要です。

サービスオペレータ
: サービスオペレータは、システムの任意のサービスインスタンスを有効または無効にしたり、その restart メソッドや refresh メソッドの実行を要求したりできます。これは、`solaris.smf.manage` 承認と `solaris.smf.modify.framework` 承認に対応します。

  サイトでは、必要に応じてカスタマイズした追加の権利プロファイルを定義することもできます。

リモートリポジトリの変更

リモートリポジトリサーバーは、追加の特権確認の結果、変更要求を拒否することがあります。「注意事項」を参照してください。

使用例

例 1　ユーザーが root にならなくても `system/cron` サービスを変更できるようにします。

次の行を `/etc/user_attr` に追加すると、ユーザー「johndoe」は root にならなくても `system/cron` サービスの再起動、有効化、無効化、またはその他の状態変更を行えるようになります。

```
johndoe::::auths=solaris.smf.manage.cron
```

例 2　ユーザーが root にならなくてもいずれかのサービスのプロパティーを変更したり、`system/cron` サービスを変更したりできるようにします。

次の行を `/etc/user_attr` に追加すると、ユーザー「janedoe」は root にならなくてもいずれかのサービスのプロパティーを変更したり、`system/cron` サービスの再起動、有効化、無効化、またはその他の状態変更を行なったりできるようになります。

```
janedoe::::auths=solaris.smf.modify,solaris.smf.manage.cron
```

関連項目 auths(1), profiles(1), svccfg(1M), prof_attr(4), user_attr(4), rbac(5), smf(5)

pkg(1)

注意事項 smf(5) の現在のバージョンでは、リモートリポジトリはサポートされていません。

サービスが limit_privileges とは異なる特権で root ユーザーとして起動するように構成されている場合、結果として得られるプロセスは特権に対応したものになります。これは、seteuid(<non-zero UID>) によって特権が basic 以下に下がると考えている開発者には予期しないことです。

名前

smf_template – サービス管理フレームワークのサービスメタデータのサポート

機能説明

テンプレートはサービス開発者によって定義され、サービスに関するメタデータを記述します。サービスの全体的な構成プロパティーや個別の構成プロパティーに記述され、人間が読める形式の説明と、有効な構成の定義を含んでいます。

管理者はSMFコマンドを介してテンプレートにアクセスし、構成値を記述したり、テンプレートと照合して構成を検証したりできます。

ツール開発者はテンプレートを使用して、サービスの構成に役立つユーザーインタフェースを作成できます。

テンプレートデータ

サービスのメタデータは、テンプレートでサービスマニフェストの一部として定義されます。

テンプレートデータの使用

`svcs -lv` コマンドと `svccfg describe` コマンドを使用すると、人間が読める形式でプロパティーに関するメタデータにアクセスできます。

svccfg(1M) の `validate` サブコマンドを使用すると、テンプレートデータと照合してサービスインスタンスやマニフェストを検証できます。テンプレートデータにアクセスするには、一連の libscf(3LIB) インタフェースを使用できます。

テンプレートの定義

テンプレートを定義するための唯一のインタフェースはサービスマニフェストです。

サービスの作成者は、サービス固有のプロパティーグループおよびそれらに導入されるプロパティーに、`common_names`、`descriptions`、`choices`、`constraints` などのテンプレートメタデータを指定するようにしてください。サービスの作成者は、少なくともプロパティーグループとプロパティーの説明をCロケールで指定する必要があります。サービスの作成者は、メソッドや依存関係といったフレームワークによって提供されるプロパティーグループには、テンプレートメタデータを指定しないようにしてください。

サービスのテンプレート定義の作成例については、「使用例」のセクションを参照してください。

テンプレートの構成

すべてのテンプレートインタフェースは、プロパティーグループに関するテンプレートデータを、最初にインスタンスで、次にサービスで、3番目にサービスのリスタータで、最後にグローバルに検索します。

プロパティーグループのテンプレートはその作成者によって定義され、特定のインスタンス、サービスとそのすべてのインスタンス、リスタータの委任、あるいはグローバルに適用されます。通常、サービスの作成者はインスタンスまたはサービスにテンプレートを定義します。インスタンスに定義されたテンプレートは、そのインスタンスだけに適用され、サービスに定義されているそのプロパティーグループのテンプレートよりも優先されます。サービスに定義されたテンプレートは、そのサービスのすべてのインスタンスに適用されます。

リスタータの作成者は、そのリスタータを使用するすべてのサービスに適用されるテンプレートをマニフェストで定義できます。これは「委任」とも呼ばれます。SMF フレームワークの作成者は、SMF フレームワーク全体に対する既知の意味を指定したプロパティーグループのテンプレートを `svc:/system/svc/global` のマニフェストに定義してあります。

グローバルにまたはリスタータによって定義されたテンプレートは、サービスまたはインスタンスで再定義されると、検証エラーとしてフラグが設定されます。サービスの作成者は、SMF フレームワークでは使用されない、そのサービスに固有のプロパティーグループだけにテンプレートを作成することで、このようなエラーを避けることができます。

プロパティーグループのテンプレートは、名前やタイプによるワイルドカード方式でも指定できます。プロパティーグループに適用可能なテンプレート定義のうち、もっとも限定的なものだけが適用されます。

テンプレートの詳細

サービスおよびインスタンスのテンプレート

`template` 要素は、template ブロックの開始を定義します。後述するほかの定義はすべて、template ブロックに含めることができます。`template` 要素は、`service` 要素または `instance` 要素に含めることができます。`service` 要素に含めた場合は、サービスおよびそのサービスのすべてのインスタンスに適用されます。`instance` 要素に含めた場合は、サービスのそのインスタンスだけに適用されます。

可能な場合は常に、サービス全体のテンプレートデータを定義することをお勧めします。

```
<service ... >
   <template>
   </template>
</service>
```

サービスおよびインスタンスの一般的な名前

サービス全体またはインスタンスには、そのサービスまたはインスタンスの目的を表す一般的な名前を定義できます。

```
<template>
  <common_name>
      <loctext xml:lang='C'>console login</loctext>
  </common_name>
<template>
```

common_name は自由形式の文字列ですが、GUI または CLI でラベルとして使用されます。

一般的な名前を定義するときは、次のガイドラインに従ってください。

- 短くします。通常は、1 語または 2 語が適切です。名前の長さは 40 文字以下にしてください。
- 明確にします。サービス名、プロパティーグループ名、またはプロパティー名は人間にはわかりにくい場合がありますが、*common_name* はその目的を明確に示すようにしてください。
- 句読文字は使用しません。*common_name* は文章ではありません。節や句を含むべきではありません。句読文字は、商標で必要になる場合にかぎり使用するようにしてください。
- 大文字は頭字語または固有名だけに使用する必要があります。英語以外のロケールでは、文の要素に適切な大文字を使用してください。

サービスおよびインスタンスの説明

description 要素には、ステータス行やツールヒントに適した、プロパティーグループに関する長い説明が記述されます。

```
<template>
  <description>
     <loctext xml:lang='C'>Provide the text login prompt on console.
     </loctext>
  </description>
<template>
```

description のガイドライン

- 正しい文法を使用します。*description* は人間が読むための文です。
- 短くします。通常は、数個の文がもっとも適切です。

ドキュメント

このサービスのドキュメントを明示的に定義することにより、サービスで問題が発生した場合やサービスの利用者が詳細な情報を必要としている場合に、ドキュメントを簡単に見つけることができます。

プロパティーグループ

`pg_pattern` 要素には、プロパティーグループの定義が記述されます。

```
<template>
  <pg_pattern name="pgname" type="pgtype" target="this" required="true">
  </pg_pattern>
</template>
```

name はプロパティーグループの名前、*type* はプロパティーグループのタイプです。

`target` は、この定義の対象を指定します。`"this"` と指定すると、これを定義しているサービスまたはインスタンスを指します。`"instance"` は、サービスの template ブロックでのみ使用でき、定義がこのサービスのすべてのインスタンスに適用される

ことを意味します。"delegate" は、リスタータの template ブロックでのみ使用でき、そのリスタータに委任されたすべてのインスタンスに適用されることを意味します。"all" は、マスターリスタータでのみ使用でき、システム上のすべてのサービスを指します。target のデフォルト値は "this" です。

required は、このプロパティーグループが必須かどうかを示します。*required* のデフォルト値は false です。*required* が true の場合は、*name* と *type* の両方を指定する必要があります。

name と *type* は、一方または両方を省略できます。これらの属性のいずれかが省略されている場合、その属性はワイルドカードとして扱われます。たとえば、pg_pattern 定義で name 属性が省略されている場合、pg_pattern は指定されたタイプのすべてのプロパティーグループに適用されます。

プロパティーグループの名前

common_name 要素には、人間が読めるローカライズされた形式で、プロパティーグループの名前が記述されます。

```
<pg_pattern ...>
  <common_name>
    <loctext xml:lang='C'>start method</loctext>
  </common_name>
</pg_pattern>
```

common_name は自由形式の文字列ですが、GUI または CLI でラベルとして使用されます。

前述の「サービスおよびインスタンスの一般的な名前」に示されている *common_name* に関するガイドラインを参照してください。

プロパティーグループの説明

description 要素には、ステータス行やツールヒントに適した、プロパティーグループに関する長い説明が記述されます。

```
<pg_pattern ...>
  <description>
    <loctext xml:lang='C'>A required method which starts the service.
   </loctext>
   </description>
</pg_pattern>
```

前述の「サービスおよびインスタンスの説明」に示されている *description* の指定方法に関するガイドラインを参照してください。

プロパティー

prop_pattern 要素には、特定のプロパティーの定義が記述されます。

```
<pg_pattern ...>
  <prop_pattern name="propname" type="proptype" required="true">
  </prop_pattern>
</pg_pattern>
```

name はプロパティーの名前、*type* はプロパティーのタイプです。

`required` は、このプロパティーが必須かどうかを示します。`required` のデフォルト値は `false` です。

name は常に必須です。`required` が `false` の場合のみ、*type* は省略可能です。

プロパティーの名前

common_name 要素には、人間が読めるローカライズされた形式で、プロパティーの名前が記述されます。

common_name は自由形式の文字列フィールドですが、GUI または CLI でラベルとして使用されます。

```
<prop_pattern ...>
<common_name>
  <loctext xml:lang='C'>retry interval</loctext>
</common_name>
</prop_pattern>
```

前述の「サービスおよびインスタンスの一般的な名前」に示されている *common_name* に関するガイドラインを参照してください。

プロパティーの単位

units 要素には、人間が読めるローカライズされた形式で、数値プロパティーの単位が記述されます。

```
<prop_pattern ...>
  <units>
    <loctext xml:lang='C'>seconds</loctext>
  </units>
</prop_pattern>
```

units のガイドライン

- 短くします。1 語または 1 ラベルだけを使用するようにします。通常は、複数形がもっとも適切です。
- 句読文字は使用しません。*units* は文章ではありません。節や句を含むべきではありません。句読文字は、商標で必要になる場合にかぎり使用するようにしてください。

プロパティーの説明

description 要素には、ステータス行やツールヒントに適した、プロパティーに関する長い説明が記述されます。

```
<prop_pattern ...>
   <description> <loctext xml:lang='C'>
    The number of seconds to wait before retry.
    </loctext> </description>
</prop_pattern>
```

前述の「サービスおよびインスタンスの説明」に示されている *description* の指定方法に関するガイドラインを参照してください。

プロパティーの可視設定

visibility 要素は、より上位のソフトウェアの簡易表示でこのプロパティーを表示するかどうかを指定します。

```
<prop_pattern ...>
  <visibility value="hidden | readonly | readwrite"/>
</prop_pattern>
```

一部のプロパティーは、内部の実装の詳細であり、構成の設定として表示されるべきではありません。単に読み取り専用のものもあります。このプロパティーは、このような制約を指定するために使用されます。値 hidden は、そのプロパティーを表示しないことを示します。`readonly` は、そのプロパティーを変更できないことを意味します。`readwrite` は、そのプロパティーが変更可能であることを示します。

これはセキュリティーメカニズムでなく、単にユーザー自身の誤りを防ぎ、CLI の出力や GUI の表示から不要な情報を取り除くためのものです。非表示のプロパティーは、多くのコマンドおよび UI の完全開示モードで表示されます。

プロパティーの形式

cardinality 要素と *internal_separators* 要素には、プロパティーの構造が記述されます。

```
<prop_pattern ...>
  <cardinality min="1" max="1"/>
  <internal_separators>,</internal_separators>
</prop_pattern>
```

cardinality は、指定できるプロパティー値の数を示します。*min* は最小数、*max* は最大数です。どちらも省略可能です。どちらも指定されていない場合、`<cardinality/>` はデフォルトと同じ、0 個以上の値になります。

internal_separators は、実際の値を複数保持するプロパティー値に使用する区切り文字を指定します。

値の制約

constraints 要素は、プロパティーに使用できる値を指定します。

```
<prop_pattern ...>
<constraints>
       <value name="blue" />
```

```
        <range min="1" max="7"/>
        <include_values type="values"/>
</constraints>
</prop_pattern>
```

value 要素には、プロパティーの取り得る値が記述されます。range には、整数の範囲が記述されます。

value と *range* は、使用を制限すると有効な説明の多くが使用できなくなるため、任意の組み合わせで使用できます。値の制約が指定されていない場合、そのプロパティーは任意の値を取ることができます。

include_values には、values ブロックで指定されたすべての値が記述されます (「値の説明」のセクションを参照)。

値の選択肢

choices ブロックは、UI でユーザーに提供する値を示します。

```
<prop_pattern ...>
<choices>
      <range min="1" max="3"/>
      <value name="vt100" />
      <value name="xterm" />
      <include_values type="constraints"/>
      <include_values type="values"/>
</choices>
</prop_pattern>
```

constraints と同様に、range と value には範囲と個々の値が記述されます。

include_values には、constraints ブロックまたは values ブロックで指定されたすべての値が記述されます (次のセクションを参照)。

値の説明

プロパティー名と同様に、プロパティーが取る値にも、わかりにくい表現のものが存在します。values 要素には、人間が読めるローカライズされた形式で、特定のプロパティー値の説明が記述されます。

```
<prop_pattern>
<values>
      <value name="blue">
              <common_name>
                      <loctext xml:lang='C'>blue</loctext>
              </common_name>
              <description>
                      <loctext xml:lang='C>
                              The color between green and indigo.
                      </loctext>
```

```
                </description>
        </value>
</values>
</prop_pattern>
```

common_name は自由形式の文字列フィールドですが、GUI または CLI でラベルとして使用されます。

前述の「サービスおよびインスタンスの一般的な名前」に示されている *common_name* に関するガイドラインを参照してください。

使用例

次のような基本的なサービスがあり、その基本的なテンプレートデータを定義する場合を考えます。

```
<?xml version="1.0"?
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='FOOfoo:foo'>
<service name='system/foo' type='service' version='1'>
        <dependency>
                name='multi-user'
                type='service'
                grouping='require_all'
                restart_on='none'
                <service_fmri value='svc:/milestone/multi-user' />
        </dependency>
        <exec_method
                type='method'
                name='start'
                exec='/opt/foo/food'
                timeout_seconds='60'>
        </exec_method>
        <exec_method
                type='method'
                name='stop'
                exec=':kill'
                timeout_seconds='60'>
        </exec_method>
        <property_group name='config' type='application'>
                <propval name='local_only' type='boolean' value='false' />
                <propval name='config_file' type='astring'
                    value='/opt/foo/foo.conf' />
        <property name='modules' type='astring'>
           <astring_list>
                 <value_node value='bar'/>
                 <value_node value='baz'/>
           </astring_list>
          </property>
        </property_group>
```

```
      <instance name='default' enabled='false' />
</service>
</service_bundle>
```

このサービスを使用する管理者に役立つように、基本的なテンプレートデータを <service> タグの内側に定義できます。もっとも役立つのは、サービス自体の目的と、サービス固有の構成を記述することです。

```
<template>
       <common_name> <loctext xml:lang='C'>
           all-purpose demonstration
       </loctext> </common_name>
       <documentation>
                <manpage title='food' section='1M'
                         manpath='/opt/foo/man' />
       </documentation>

       <pg_pattern name='config' type='application' target='this'
           required='true'>
               <description> <loctext xml:lang='C'>
                   Basic configuration for foo.
                </loctext> </description>
                <prop_pattern name='local_only' type='boolean'
                    required='false'>
                      <description> <loctext xml:lang='C'>
                          Only listen to local connection requests.
                      </loctext> </description>
                </prop_pattern>
                <prop_pattern name='config_file' type='astring'
                     required='true'>
                       <cardinality min='1' max='1'/>
                       <description> <loctext xml:lang='C'>
                            Configuration file for foo.
                        </loctext> </description>
                 </prop_pattern>
                 <prop_pattern name='modules' type='astring'
                     required='false'>
                         <description> <loctext xml:lang='C'>
                             Plugin modules for foo.
                          </loctext> /description>
                          <values>
                                   <value name='bar'>
                                   <description> <loctext xml:lang='C'>
                                       Allow foo to access the bar.
                                    </loctext> </description>
                                    </value>
                                    <value name='baz'>
                                    <description> <loctext xml:lang='C'>
                                       Allow foo to access baz functions.
```

```
                                        </loctext> </description>
                                        </value>
                                        <value name='qux'>
                                        <description> <loctext xml:lang='C'>
                                           Allow foo to access qux functions.
                                         </loctext> </description>
                                         </value>
                                </values>
                                <choices>
                                    <include_values type='values'/>
                                </choices>
                        <prop_pattern>
                </pg_pattern>
        </template>
```

ファイル　/usr/share/lib/xml/dtd/service_bundle.dtd.1

関連項目　svcs(1), svccfg(1M), libscf(3LIB), service_bundle(4), smf(5)

/usr/apache2/2.2/man

http://httpd.apache.org

名前

solaris10 – Solaris 10 ブランドゾーン

機能説明

`solaris10` ブランドは、`brands(5)` で説明されているブランドゾーンフレームワークを使用して、Solaris 10 バイナリアプリケーションを変更することなく、最新の Solaris オペレーティングシステムカーネルを備えたマシンで実行できるようにします。

Oracle Solaris 10 ゾーンは、Oracle Solaris 11 カーネルで動作する x86 および SPARC Solaris 10 9/10 (またはそれ以降にリリースされた Oracle Solaris 10 update) ユーザー環境をホストする solaris10 ブランドゾーンです。

元のシステムで最初にカーネルパッチ 142909-17 (SPARC) または 142910-17 (x86/x64)、あるいはそれ以降のバージョンをインストールした場合は、以前の Oracle Solaris 10 リリースを使用できます。

`solaris10` ブランドには、Solaris 10 システムイメージを非大域ゾーンにインストールするために必要なツールが含まれています。Solaris 10 ネイティブゾーンを `solaris10` ブランドゾーンに移行するために必要なツールもサポートされています。このブランドでは、最新の Solaris オペレーティングシステムが稼働している SPARC または x86 マシンのいずれかでの、32 ビットおよび 64 ビットの Solaris 10 アプリケーションの実行をサポートしています。

構成と管理

`solaris10` ブランドは完全ルート非大域ゾーンモデルをサポートします。Solaris 10 ソフトウェアの必須パッケージおよび追加パッケージのすべてが、ゾーン固有のファイルシステムにインストールされます。

ゾーンは固有の `zfs(1M)` データセット上に構築される必要があり、ZFS だけがサポートされています。ZFS データセットは、ゾーンのインストール時または接続時に自動的に作成されます。ZFS データセットを作成できない場合、ゾーンのインストールや接続は行われません。

`zonecfg(1M)` ユーティリティーは、`solaris10` ブランドゾーンを構成するために使用されます。ゾーンの作成時に `SYSsolaris10` テンプレートを使用することも、構成を手動で設定することもできます。ブランドゾーンのインストールが完了したあとは、そのゾーンのブランドの変更や削除を行うことはできません。`zoneadm(1M)` ユーティリティーは、ゾーンのブランドタイプの報告およびゾーンの管理に使用されます。`zlogin(1)` ユーティリティーは、ゾーンにログインするために使用されます。

現在、委任された ZFS データセット構成のサポートは実験的なもので、まだテストされていません。準仮想化された xVM ドメインでこれらのゾーンを実行するためのサポートは実験的なもので、ゾーン内の 64 ビット x86 アプリケーションで問題が見つかっています。`/dev/sound` デバイスをブランドゾーン内に構成することはできません。また、大域ゾーンで `mdb(1)` と `dtrace(1M)` を使用して `solaris10` ブランドゾーン内で実行されているプロセスを検査する場合、これらは完全には機能しません。

solaris10 ブランドのインストーラでは、インストール済み Solaris 10 システムのイメージからのゾーンのインストールをサポートしています。これには、完全な flash_archive(4)、cpio(1)、または pax(1) xustar アーカイブを使用できます。cpio アーカイブは gzip(1) または bzip2(1) で圧縮することもできます。このイメージとしては、レベル 0 ufsdump(1M)、または Solaris 10 システムのルートディレクトリツリーの最上位パスも使用できます。標準の Solaris 10 配布メディアからゾーンをインストールすることはできません。

ネイティブゾーンを Solaris 10 システムから最新の Solaris オペレーティングシステムカーネルに移行するするために、attach サブコマンドでは、インストール済み Solaris 10 ネイティブゾーンのアーカイブからのゾーンのインストールをサポートしています。インストーラの場合と同様に、zonepath の cpio(1) または pax(1) xustar アーカイブを使用できます。cpio アーカイブは gzip(1) または bzip2(1) で圧縮することもできます。このイメージとしては、Solaris 10 ゾーンの zonepath ディレクトリツリーの最上位パスもイメージとして使用できます。Solaris 10 ネイティブゾーンからの移行に加え、同じ移行オプションを使用して solaris10 ブランドゾーンをあるホストから別のホストに移行することもできます。Solaris 10 を移行するときに、ゾーンが「疎ルート」ゾーンとして構成されている場合があります。この場合、アーカイブを作成する前に、ホスト上でゾーンを準備するようにしてください。これにより、継承されたディレクトリがアーカイブに含まれることが保証されます。

サブコマンド

サポートされている zoneadm(1M) ブランド固有サブコマンドの引数は次のとおりです。

attach [-a *archive* | -d *path*] [-c *sysidcfg*]
: 指定された Solaris 10 ネイティブゾーンイメージをブランドゾーンに接続します。-a も -d も指定されていない場合は、ゾーンの zonepath にゾーンのファイルがすでに正しくインストールされているとみなされます。

  -a *archive*
  : インストール済み Solaris 10 ネイティブゾーンまたは solaris10 ブランドゾーンの zonepath にある、cpio(1)、pax(1) xustar、または zfs アーカイブへのパス。cpio および zfs アーカイブは、gzip または bzip2 を使用して圧縮できます。

    このオプションは、ゾーンの状態が「使用不可」の場合は使用されない可能性があります。

    警告: このオプションは Solaris の将来のリリースで削除される可能性があります。「zoneadm install -a archive」の使用をお勧めします。

  -c *sysidcfg*
  : *sysidcfg* ファイルを指定すると、接続後にゾーン上で sys-unconfig が実行されます。その後、*sysidcfg* ファイルがゾーンに適用されます。

-d *path*　インストール済み Solaris 10 ネイティブゾーンまたは solaris10 ブランドゾーンの zonepath の、zonepath ディレクトリのパス。

このオプションは、ゾーンの状態が「使用不可」の場合は使用されない可能性があります。

警告: このオプションは Solaris の将来のリリースで削除される可能性があります。「zoneadm install -a archive」の使用をお勧めします。

clone [-c sysidcfg]
既存のインストール済みゾーンをコピーすることでゾーンのインストールを行います。このサブコマンドは、ゾーンをインストールするための代替手段となります。

-c sysidcfg　複製されたゾーンの構成解除後に適用する *sysidcfg* ファイルを指定します。

install [-a *archive*] [-d *path*] [-p] [-s] [-u] [-v] \
[-c sysidcfg]
指定された Solaris 10 システムイメージをゾーンにインストールします。-u オプションまたは -p オプションのどちらかが必須で、-a オプションまたは - d オプションのどちらかが必須です。

-a *archive*
インストールされた Solaris 10 システム、インストールされた Solaris 10 ネイティブゾーン、または solaris10 ブランドゾーンの flash_archive(4)、cpio(1)、pax(1) xustar、zfs アーカイブ、またはレベル 0 の ufsdump(1M) へのパス。cpio および zfs アーカイブは、gzip または bzip2 を使用して圧縮できます。

-c *sysidcfg*
インストール後に適用する *sysidcfg* ファイルを指定します。

-d *path*
インストール済み Solaris 10 システムのルートディレクトリのパス。

-p
ゾーンのインストール後にシステム構成を維持します。

-s
サイレントインストールします。

-u
ゾーンのインストール後にゾーンに対して sys-unconfig を実行します。

-v
インストールプロセスの詳細情報を出力します。

アプリケーションのサポート

solaris10 ゾーンはユーザーレベルの Solaris 10 アプリケーションだけをサポートします。Solaris 10 デバイスドライバや Solaris 10 カーネルモジュールを solaris10 ゾーン内から使用することはできません。ただし、カーネルモジュールによっては、最新バージョンの Solaris カーネルモジュールをユーザーレベルの Solaris 10 アプリケーションに使用できる場合があります。

使用例

例 1 インストール用の ZFS フラッシュアーカイブを作成する

次の例は、物理から仮想 (P2V) 移行用のアーカイブを作成する方法を示しています。これは、Solaris 10 が動作しているシステムの大域ゾーンで実行されます。Solaris 10 システムに、構成済み、インストール済み、または稼働中の非大域ゾーンがあってはいけません。Solaris 10 システムでは、そのルートファイルシステムとして ZFS または UFS を使用できます。

```
# flarcreate -n s10box -c /net/somehost/p2v/s10box.flar
```

例 2 フラッシュアーカイブを使用して solaris10 ブランドゾーンをインストールする

次の例では、例 1 からのアーカイブを使用してゾーンをインストールします。このゾーンは zonecfg(1M) を使用して構成済みであり、brand プロパティーが solaris10 に設定されているものと想定します。

```
# zoneadm -z s10p2v install -a /net/somehost/p2v/s10box.flar -p
```

例 3 接続用の ZFS アーカイブを作成する

次の例は、仮想から仮想 (V2V) 移行用のアーカイブを作成する方法を示しています。solaris10 ブランドゾーンの zonepath は /zones/v2vzone であると想定します。

まず、zonepath dataset の名前を決定します。

```
# dataset=$(zfs list -H -o name /zones/v2vzone)
```

次に、ゾーンのデータセットのスナップショットを作成します。

```
# zfs snapshot -r $dataset@v2v
```

最後に、bzip2 で圧縮された ZFS 内蔵式の再帰的ストリームを生成します。

```
# zfs send -rc $dataset@v2v | bzip2 > /net/somehost/v2v/v2v.zfs.bz2
```

例 4 ZFS アーカイブを使用してゾーンを接続する

次の例では、ZFS アーカイブを使用してゾーンを接続します。このゾーンは zonecfg(1M) を使用して構成済みであり、brand プロパティーが solaris10 に設定されているものと想定します。

```
# zoneadm -z v2vzone attach -a /net/somehost/v2v/v2v.zfs.bz2
```

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones/brand/brand-solaris10
インタフェースの安定性	確実

関連項目

cpio(1), mdb(1), pax(1), zlogin(1), dtrace(1M), ufsdump(1M), zfs(1M), zoneadm(1M), zonecfg(1M), flash_archive(4), attributes(5), brands(5), zones(5)

名前

solaris, ipkg – solaris ブランドゾーン

機能説明

`solaris` ブランドは、`brands(5)` で説明されているブランドゾーンフレームワークを使用して、大域ゾーンにインストールされているものと同じソフトウェアがインストールされたゾーンを実行します。`solaris` ブランドを使用するときは、大域ゾーンとの間でシステムソフトウェアの同期が常に保たれている必要があります。ゾーン内のシステムソフトウェアパッケージは、イメージパッケージングシステムを使用して管理されます。`pkg(5)` を参照してください。

構成と管理

`solaris` ブランドは完全ルート非大域ゾーンモデルをサポートします。システムソフトウェアの必須パッケージおよび追加パッケージのすべてが、ゾーン固有のファイルシステムにインストールされます。ゾーンは固有の `zfs(1M)` データセット上に構築される必要があり、ZFS だけがサポートされています。ZFS データセットは、ゾーンのインストール時または接続時に自動的に作成されます。ZFS データセットを作成できない場合、ゾーンのインストールや接続は行われません。

サブコマンド

`zoneadm(1M)` でサポートされている `solaris` ブランド固有サブコマンドのオプションは次のとおりです。

`attach` [`-a` *archive* | `-d` *path* | `-z` *ZBE*] [`-u` | `-U`] [`-c` *config_profile* `.xml` | *dir*]

指定された `solaris` ブランドゾーンイメージをゾーンに接続します。`-a` も `-d` も指定されていない場合は、ゾーンのゾーンパスにゾーンのファイルがすでに正しくインストールされているとみなされます。`zoneadm` は、ゾーンの接続先となるマシンのパッケージのレベルを確認します。ゾーンが依存する大域ゾーンのパッケージがソースマシン上の依存パッケージと異なっている (リビジョン番号が異なっている) 場合、`zoneadm` はこれらの競合を報告し、接続を実行しません。

接続先のシステムにある依存パッケージの方がソースシステムのものよりすべて新しい (リビジョン番号が大きい) 場合は、依存パッケージを `-u` または `-U` オプションで更新して、新しいシステムにあるパッケージのリビジョンに一致させることができます。

ゾーンを接続する際、複数のゾーンブート環境 (ZBE) を存在させることが可能であり、`attach` サブコマンドがどれを接続するかを決定します。選択条件は次のとおりです。最初に一致したものが使用されます。

- `-z` オプションを使用して ZBE が指定された場合、それが選択されます。
- ZBE が 1 つしかない場合、それが選択されます。
- 大域ゾーンブート環境に関連付けられた `ZBE` が 1 つしかない場合、それが選択されます。
- 大域ゾーンブート環境に関連付けられたアクティブな `ZBE` が 1 つしかない場合、それが選択されます。
- アクティブな `ZBE` が 1 つしかない場合、それが選択されます。

選択された ZBE が別の大域ブート環境に関連付けられている場合、ZBE のクローンが作成され、選択された ZBE のクローンが接続されます。ブート環境に関する詳細情報については、beadm(1M) を参照してください。

-a *archive*
警告: このオプションは Solaris の将来のリリースで削除される可能性があります。「zoneadm install -a archive」の使用をお勧めします。

インストールされた Oracle Solaris ブランドゾーンの cpio(1)、pax(1) xustar、または zfs(1M) アーカイブへのパス。

cpio および pax アーカイブは、ゾーンパスまたはゾーンルートのものである必要があり、絶対パスを含めてはいけません (パスが / で始まってはいけない)。ZFS アーカイブは、zonepath dataset、zone rpool dataset、または単一のゾーンブート環境のものにできます。

ZFS アーカイブに複数の ZBE が含まれている場合、-z オプションを使用して特定の ZBE を接続する必要があることを示すメッセージが表示され、接続が失敗することがあります。このような場合、アーカイブからの ZBE は抽出されたままになり、抽出された ZBE のうちの 1 つが -z オプションを使用して接続されるまでは、このゾーンでこれ以上 -a および -d オプションを使用しても無効になります。抽出された ZBE のうちの 1 つを接続するのではなく、抽出された ZBE を削除するのが望ましい場合は、zoneadm -z <zone> mark incomplete のあとに zoneadm -z <zone> uninstall を使用します。

cpio および ZFS アーカイブは、gzip または bzip2 で圧縮できます。

このオプションは、ゾーンの状態が「使用不可」の場合は使用されない可能性があります。

-c *config_profile*.xml | *dir*
リポジトリからのインストール後に適用するプロファイルまたはプロファイルのディレクトリを指定します。

すべてのプロファイルは、.xml 拡張子を持つ必要があります。

-d *path*
警告: このオプションは Solaris の将来のリリースで削除される可能性があります。「zoneadm install -a archive」の使用をお勧めします。

solaris ブランドゾーンのゾーンパスのゾーンパスディレクトリへのパス。

このオプションは、ゾーンの状態が「使用不可」の場合は使用されない可能性があります。

-u
ゾーン内にある最小限の数のパッケージを更新して、大域ゾーンにインストールされているパッケージとの互換性をゾーンのパッケージに持たせます。

-U
ゾーン内にあるすべてのパッケージを、大域ゾーンにインストールされているパッケージと互換性のある最新のバージョンに更新します。

-z *ZBE*
指定された既存のゾーンブート環境を接続します。指定されたゾーンブート環境が異なる大域ゾーンに関連付けられている場合は、指定された ZBE のクローンが作成され、ZBE のクローンが接続されます。

clone [-c *config_profile*.xml | *dir*]

-c *config_profile*.xml | *dir*
リポジトリからのインストール後に適用するプロファイルまたはプロファイルのディレクトリを指定します。

すべてのプロファイルは、.xml 拡張子を持つ必要があります。

install [-m *manifest.xml*] [-c *config_profile*.xml | *dir*]
install [-a *archive* | -d *path*] [-U] [-p] [-s] [-u] [-v] [-c *config_profile*.xml | *dir*]
solaris ブランドのインストーラでは、ゾーンをソフトウェアリポジトリからインストールすることも、同じリリースが稼働しているインストール済みシステムのイメージからインストールすることもサポートしています。これは、cpio(1)、pax(1) xustar、または ZFS アーカイブにできます。cpio または ZFS アーカイブは、gzip または bzip2 で圧縮できます。また、イメージとして、システムのルートツリーの最上位パス、または既存のゾーンパスも使用できます。

-a オプションも -d オプションも指定されていない場合、ゾーンはリポジトリからインストールされます。ゾーン内の追加のパッケージをインストールする場合、デフォルトのゾーンマニフェスト /usr/share/auto_install/manifest/zone_default.xml をコピーして、必要なパッケージを含むように編集することができます。この変更されたマニフェストは、-m オプションを付けてインストールするように指定してください。

ゾーンをシステムまたはゾーンイメージからインストールするには、-a または -d オプションのどちらかが必要です。必要に応じて、イメージのアクティブな ZBE 内のソフトウェアは、大域ゾーンのパッケージとの互換性を保つために必要な最小限の変更で更新されます。-U オプションが指定されている場合は、イメージのアクティブな ZBE 内のすべてのソフトウェアが、アクティブな ZBE と互換性がある最新バージョンに更新されます。-a または -d オプションのどちらかが使用されている場合は、-u または -p オプションのどちらかも必要になります。

-a *archive*
インストールされた大域ゾーンまたは非大域ゾーンの cpio(1)、pax(1) xustar、または ZFS アーカイブへのパス。

ZFS アーカイブに複数のブート環境が含まれている場合は、アクティブなブート環境がインストールされます。どのブート環境がアクティブなブート環境なのかをインストールが決定できない場合、インストールは抽出されたブート環境の一覧を表示し、-z オプションを使用した attach コマンドで特定のブート環境を接続するように提案します。

cpio および ZFS アーカイブは、gzip または bzip2 で圧縮できます。

-c *config_profile*.xml | *dir*
: リポジトリからのインストール後に適用するプロファイルまたはプロファイルのディレクトリを指定します。

すべてのプロファイルは、.xml 拡張子を持つ必要があります。

-d *path*
: solaris ブランドゾーンのゾーンパスまたは Solaris 11 大域ゾーンのルートディレクトリのゾーンパスディレクトリへのパス。

-m *manifest.xml*
: 自動インストーラに対して指定されるマニフェストファイル。

-p
: アーカイブまたはパスからゾーンをインストールしたあと、システム構成を保持します。

-s
: サイレントインストール

-u
: システムをインストールしたあと、構成解除します。

-U
: ゾーン内にあるすべてのパッケージを、大域ゾーンにインストールされているパッケージと互換性のある最新のバージョンに更新します。-U オプションは、-a または -d オプションのどちらかが使用されている場合にのみ使用できます。

-v
: インストールプロセスの詳細情報を出力します。

-z *ZBE*
: 指定された既存のゾーンブート環境を接続します。指定されたゾーンブート環境が異なる大域ゾーンに関連付けられている場合は、指定された ZBE のクローンが作成され、ZBE のクローンが接続されます。

使用例

例 1　P2V の ZFS アーカイブの作成

次の例では、物理から仮想 (P2V) 移行用のアーカイブを作成する方法を示しています。これは、構成済み、インストール済み、または稼働中の非大域ゾーンがないシステムの大域ゾーンで実行されます。ルートプールは rpool という名前であると仮定しています。

まず、ルートプール全体のスナップショットを作成します。

```
# zfs snapshot -r rpool@p2v
```

次に、対象システムでは必要にならないため、スワップおよびダンプデバイスに関連したスナップショットを破棄します。

```
# zfs destroy rpool/swap@p2v
# zfs destroy rpool/dump@p2v
```

最後に、gzip で圧縮された ZFS レプリケーションストリームアーカイブを生成します。この例では、これはリモート NFS サーバーに格納されています。

```
# zfs send -R rpool@p2v | gzip > /net/somehost/p2v/p2v.zfs.gz
```

例 2　V2V の ZFS アーカイブの作成

次の例では、仮想から仮想 (V2V) 移行用のアーカイブを作成する方法を示しています。ゾーンの zonepath は /zones/v2vzone と仮定しています。

まず、zonepath データセットの名前を決定します。

```
# dataset=$(zfs list -H -o name /zones/v2vzone)
```

次に、ゾーンのデータセットのスナップショットを作成します。

```
# zfs snapshot -r $dataset@v2v
```

最後に、bzip2 で圧縮された ZFS 内蔵式の再帰的ストリームを生成します。

```
# zfs send -rc $dataset@v2v | bzip2 > /net/somehost/v2v/v2v.zfs.bz2
```

例 3　ZFS アーカイブを使用してゾーンをインストールする

次の例では、ZFS アーカイブを使用してゾーンをインストールします。

```
# zoneadm -z v2vzone install -p -a /net/somehost/v2v/v2v.zfs.bz2
```

例 1 で作成された P2V アーカイブのインストールにも同様のコマンドを使用できます。

属性

属性についての詳細は、attributes(5) を参照してください。

属性タイプ	属性値
使用条件	system/zones
インタフェースの安定性	不確実

関連項目

cpio(1), pax(1), beadm(1M), sysconfig(1M), zfs(1M), zlogin(1), zonename(1), zoneadm(1M), zonecfg(1M), attributes(5), brands(5), privileges(5), zones(5)

pkg(5)、IPS 統合で利用可能

名前

標準, ANSI, C, C++, ISO, POSIX, POSIX.1, POSIX.2, SUS, SUSv2, SUSv3, SVID, SVID3, XNS, XNS4, XNS5, XPG, XPG3, XPG4, XPG4v2 – Oracle Solaris でサポートされている標準および仕様

機能説明

Oracle Solaris は、IEEE Std 1003.1 および IEEE Std 1003.2 (一般には、それぞれ POSIX.1 および POSIX.2 と呼ばれます) をサポートしています。次の表は、これらの標準の各バージョンと簡単な説明、およびそれに最初に準拠した SunOS または Solaris リリースを示しています。

POSIX 標準	説明	解放
POSIX.1-1988	システムインタフェースおよびヘッダー	SunOS 4.1
POSIX.1-1990	POSIX.1-1988 更新	Solaris 2.0
POSIX.1b-1993	リアルタイム拡張	Solaris 2.4
POSIX.1c-1996	スレッド拡張	Solaris 2.6
POSIX.2-1992	シェルおよびユーティリティー	Solaris 2.5
POSIX.2a-1992	対話型シェルおよびユーティリティー	Solaris 2.5
POSIX.1–2001	POSIX.1-1990、POSIX.1b-1993、POSIX.1c-1996、POSIX.2-1992、および POSIX.2a-1992 更新	Solaris 10
POSIX.1-2004	POSIX.1-2001 更新	Oracle Solaris 11
POSIX.1-2008	POSIX.1-2004 更新	Oracle Solaris 11

Oracle Solaris はまた、X/Open Common Applications Environment (CAE) Portability Guide Issue 3 (XPG3) および Issue 4 (XPG4)、Single UNIX Specification (SUS、XPG4v2 ともいう)、Single UNIX Specification, Version 2 (SUSv2)、および Single UNIX Specification, Version 3 (SUSv3) もサポートしています。XPG4 と SUS のどちらにも、Networking Services Issue 4 (XNS4) が含まれています。SUSv2 には、Networking Services Issue 5 (XNS5) が含まれています。

次の表は、各 X/Open 仕様と簡単な説明、およびそれに最初に準拠した SunOS または Solaris リリースを示しています。

X/Open CAE 仕様	説明	解放
XPG3	SVID3 のユーティリティーを含む POSIX.1-1988 のスーパーセット	SunOS 4.1

X/Open CAE 仕様	説明	解放
XPG4	XPG3 からの POSIX 標準の拡張を含む POSIX.1-1990、POSIX.2-1992、および POSIX.2a-1992 のスーパーセット	Solaris 2.4
SUS (XPG4v2)	一般的なアプリケーションパッケージで広範に使用されている履歴 BSD インタフェースを含む XPG4 のスーパーセット	Solaris 2.6
XNS4	ソケットおよび XTI インタフェース	Solaris 2.6
SUSv2	POSIX.1b-1993、POSIX.1c-1996、および ISO/IEC 9899 (C Standard) Amendment 1 をサポートするために拡張された SUS のスーパーセット	Solaris 7
XNS5	XNS4 のスーパーセットおよび LP64 クリーンの派生型。	Solaris 7
SUSv3	POSIX.1–2001 と同じ	Solaris 10

XNS4 仕様は ILP32 (32 ビット) 環境での使用に関してのみ安全であり、LP64 (64 ビット) アプリケーション環境で使用すべきではありません。ILP32 および LP64 環境にわたって移植可能な LP64 クリーンのインタフェースを備えた XNS5 または SUSv3 を使用してください。Solaris リリース 7 から Oracle Solaris 11 までは、ILP32 環境と LP64 環境の両方をサポートしています。

Solaris リリース 7 - 10 は、The Open Group の UNIX 98 Product Standard に準拠するようにブランド化されています。Solaris 10 から Oracle Solaris 11 までは、The Open Group の UNIX 03 Product Standard に準拠するようにブランド化されています。

Solaris リリース 2.0 から Oracle Solaris 11 までは、System V Interface Definition, Third Edition, Volumes 1 - 4 (SVID3) で指定されたインタフェースをサポートしています。ただし、この仕様 (UNIX Systems Laboratories) の開発者はすでに活動しておらず、またこの仕様は POSIX および X/Open CAE 仕様に従っているため、現在この仕様に準拠するには何が必要かについてはなんらかの意見の不一致があることに注意してください。

Oracle Solaris Studio 12.3 C コンパイラがインストールされている場合、Oracle Solaris 11 は ANSI X3.159-1989 Programming Language - C および ISO/IEC 9899:1990 Programming Language - C (C) インタフェースをサポートします。

Oracle Solaris Studio 12.3 C コンパイラがインストールされている場合、Oracle Solaris 11 は ISO/IEC 9899:1990 Amendment 1:1995: C Integrity をサポートします。

Oracle Solaris Studio 12.3 C コンパイラがインストールされている場合、Oracle Solaris 11 は ISO/IEC 9899:1999 Programming Language - C をサポートします。

Oracle Solaris Studio 12.3 C++ コンパイラがインストールされている場合、Oracle Solaris 11 は ISO/IEC 14882:1998 Programming Language - C++ をサポートします。その標準のサポートされていない機能は、コンパイラの README ファイルで説明されています。

ユーティリティー

POSIX.2、POSIX.2a、XPG4、SUS、または SUSv2 に必要な動作が Solaris ユーティリティーの履歴動作と競合する場合、そのユーティリティーの元の Solaris バージョンは変更されていません。標準に準拠した新しいバージョンは `/usr/xpg4/bin` で提供されています。POSIX.1–2001 または SUSv3 に必要な動作が Solaris ユーティリティーの履歴動作と競合する場合、標準に準拠した新しいバージョンは `/usr/xpg4/bin` または `/usr/xpg6/bin` で提供されています。POSIX.1–2001 または SUSv3 に必要な動作が POSIX.2、POSIX.2a、SUS、または SUSv2 と競合する場合、SUSv3 標準に準拠した新しいバージョンは `/usr/xpg6/bin` で提供されています。

標準に準拠したユーティリティーを使用するアプリケーションは、適切なユーティリティーを取得するために、`PATH` (sh(1) または ksh(1)) または `path` (csh(1)) 環境変数を設定して、次に示されているディレクトリを指定されている順序で指定する必要があります:

SVID3、XPG3

1. `/usr/bin`
2. 使用しているコンパイラのバイナリを含むディレクトリ
3. アプリケーションで必要なバイナリを含むほかのディレクトリ

POSIX.2、POSIX.2a、SUS、SUSv2、XPG4

1. `/usr/xpg4/bin`
2. `/usr/bin`
3. 使用しているコンパイラのバイナリを含むディレクトリ
4. アプリケーションで必要なバイナリを含むほかのディレクトリ

POSIX.1–2001、SUSv3

1. `/usr/xpg6/bin`
2. `/usr/xpg4/bin`
3. `/usr/bin`
4. 使用しているコンパイラのバイナリを含むディレクトリ
5. アプリケーションで必要なバイナリを含むほかのディレクトリ

アプリケーションが `execlp()` または `execvp()` (exec(2) を参照) を使用してシェルファイルを実行するか、または system(3C) を使用する場合、そのシェルファイルを解釈するために使用されるシェルは呼び出し元が準拠する標準によって異なります:

標準	使用されるシェル
1989 ANSI C、1990 ISO C、1999 ISO C、POSIX.1 (1990?2001)、SUS、SUSv2、SUSv3、XPG4	/usr/xpg4/bin/sh
POSIX.1 (1988)、SVID3、XPG3 (標準が指定されていない)	/usr/bin/sh

機能テストマクロ

機能テストマクロは、C 標準で指定されている機能セット以外で求められる追加の機能セットを示すために、アプリケーションによって使用されます。アプリケーションが特定の標準 (POSIX や X/Open CAE など) で定義されているインタフェースおよびヘッダーのみを使用する場合、そのアプリケーションは、その標準で指定されている適切な機能テストマクロを定義するだけで済みます。アプリケーションがその標準で定義されていないインタフェースおよびヘッダーを使用している場合は、適切な標準の機能テストマクロの定義に加えて、__EXTENSIONS__ も定義する必要があります。__EXTENSIONS__ を定義すると、アプリケーションには、指定された標準と競合していないすべてのインタフェースおよびヘッダーへのアクセスが提供されます。アプリケーションは、__EXTENSIONS__ をコンパイルのコマンド行か、またはアプリケーションのソースファイル内のどちらかで定義する必要があります。

1989 ANSI C、1990 ISO C、1999 ISO C

アプリケーションが C アプリケーションに準拠していることを示すためにどの機能テストマクロも定義する必要はありません。

ANSI/ISO C++

ANSI/ISO C++ では、機能テストマクロが定義されていません。コンパイラのデフォルト設定とコマンド行オプションに基づいてコンパイラによって事前に定義されている標準 C++ の通知マクロ __cplusplus が 199711 以上の値に設定されている場合、コンパイラは標準に準拠したモードで動作し、C++ 標準への準拠を示します。199711 の値は ISO/IEC 14882:1998 への準拠を示します (その標準に必要です)。(上で説明したように、この標準への準拠は不完全です。)

POSIX や X/Open CAE では C++ バインディングが定義されていないため、_POSIX_SOURCE、_POSIX_C_SOURCE、_XOPEN_SOURCE などの機能テストマクロを指定すると、標準 C++ とこれらの仕様の競合する要件のためにコンパイルエラーになる場合があります。

POSIX

POSIX.1 アプリケーションに準拠するように設計されたアプリケーションは、いずれかのヘッダーをインクルードする前に、この標準で指定されている機能テストマクロを定義する必要があります。次に示されている標準の場合、アプリケーションは示されている機能テストマクロを定義する必要があります。アプリケーション作成者は、対応する標準を確認して、目的のオプションが実装によってサポートされているかどうかを確認するために照会できるほかのマクロを探す必要があります。

POSIX 標準	機能テストマクロ
POSIX.1-1990	_POSIX_SOURCE
POSIX.1-1990 および POSIX.2-1992 C-Language Bindings Option	_POSIX_SOURCE および _POSIX_C_SOURCE=2
POSIX.1b-1993	_POSIX_C_SOURCE=199309L
POSIX.1c-1996	_POSIX_C_SOURCE=199506L
POSIX.1–2001	_POSIX_C_SOURCE=200112L

SVID3

SVID3 仕様は、アプリケーションが SVID3 の要件を満たすように記述されていることを示すためにどの機能テストマクロも指定しません。SVID3 仕様は、C 標準が完成する前に記述されました。

X/Open CAE

いずれかの X/Open CAE 仕様に準拠するアプリケーションを構築またはコンパイルするには、次のガイドラインに従ってください。アプリケーションに CAE 機能と POSIX 機能の両方が必要な場合、そのアプリケーションは POSIX 機能テストマクロを設定する必要はありません。

XPG3 アプリケーションは、_XOPEN_SOURCE を定義する必要があります。_XOPEN_SOURCE がある値で定義されている場合、その値は 500 未満である必要があります。

XPG4 アプリケーションは _XOPEN_SOURCE を定義し、_XOPEN_VERSION=4 を設定する必要があります。_XOPEN_SOURCE がある値で定義されている場合、その値は 500 未満である必要があります。

SUS (XPG4v2) アプリケーションは、_XOPEN_SOURCE を定義し、_XOPEN_SOURCE_EXTENDED=1 を設定する必要があります。_XOPEN_SOURCE がある値で定義されている場合、その値は 500 未満である必要があります。

SUSv2 アプリケーションは、_XOPEN_SOURCE=500 を定義する必要があります。

SUSv3 アプリケーションは、_XOPEN_SOURCE=600 を定義する必要があります。

コンパイル

Oracle Solaris Studio 12.3 C コンパイラは、ISO/IEC 99899:1999 (1999 ISO C Language) 標準に準拠したコンパイルシステムと c99 ユーティリティーを提供します。

アプリケーションをリンクするために ld が直接使用されている場合は、そのアプリケーションが POSIX.1–2001 または SUSv3 に準拠していないかぎり、いずれかの link/load コマンド行で /usr/lib/values-xpg4.o を指定する必要があります。準拠し

ている場合は、いずれかの link/load コンパイル行で /usr/lib/values-xpg6.o を指定する必要があります。アプリケーションをリンクするために cc または CC が使用されている場合、コンパイラは適切なファイルを自動的に追加します。ただし、次の表で説明されている方法でアプリケーションを構築することをお勧めします。

XNS4 または XNS5 に準拠したアプリケーションは、それぞれ SUS または SUSv2 に対して指定された機能テストマクロの定義に加えて、いずれかの link/load コマンド行で -l XNS をインクルードする必要があります。

コンパイラが redefine_extname プラグマ機能をサポートしている場合 (Oracle Solaris Studio 12.3 C コンパイラおよび Oracle Solaris Studio 12.3 C++ コンパイラは、この機能をサポートしていることを示すためにマクロ __PRAGMA_REDEFINE_EXTNAME を定義しています)、標準ヘッダーは、関数名をライブラリのエントリポイント名に正しくマップするために #pragma redefine_extname 指令を使用します。このマッピングによって、ISO C、POSIX、および X/Open の名前空間予約に対する完全なサポートが提供されます。

このプラグマ機能がコンパイラでサポートされていない場合、ヘッダーは、内部の関数名を適切なライブラリのエントリポイント名にマップするために #define 指令を使用します。この例では、lf64(5) のマニュアルページに示されている明示的な 64 ビットファイルオフセットシンボルは、代わりのエントリポイントを指定するために実装で使用されるため、アプリケーションではこれらの名前の使用を避けるようにしてください。

Oracle Solaris Studio 12.3 C コンパイラを使用している場合は、上に示されている仕様に準拠するアプリケーションを、次の表に示されているユーティリティーとフラグを使用してコンパイルするようにしてください:

Specification	Compiler/Flags	Feature Test Macros
1989 ANSI C and 1990 ISO C	c89	none
1999 ISO C	c99	none
SVID3	cc -Xt -xc99=none	none
POSIX.1-1990	c89	_POSIX_SOURCE
POSIX.1-1990 and POSIX.2-1992 C-Language Bindings Option	c89	_POSIX_SOURCE and POSIX_C_SOURCE=2
POSIX.1b-1993	c89	_POSIX_C_SOURCE=199309L
POSIX.1c-1996	c89	_POSIX_C_SOURCE=199506L
POSIX.1-2001	c99	_POSIX_C_SOURCE=200112L

POSIX.1c-1996	c89	_POSIX_C_SOURCE=199506L
CAE XPG3	cc -Xa -xc99=none	_XOPEN_SOURCE
CAE XPG4	c89	_XOPEN_SOURCE and _XOPEN_VERSION=4
SUS (CAE XPG4v2) (includes XNS4)	c89	_XOPEN_SOURCE and _XOPEN_SOURCE_EXTENDED=1
SUSv2 (includes XNS5)	c89	_XOPEN_SOURCE=500
SUSv3	c99	_XOPEN_SOURCE=600

LP64 (64 ビット) プログラミング環境をサポートしているプラットフォームの場合は、XNS5 ライブラリコールを使用している SUSv2 に準拠した LP64 アプリケーションを、次の形式のコマンド行で構築するようにしてください:

```
c89 $(getconf XBS5_LP64_OFF64_CFLAGS) -D_XOPEN_SOURCE=500 \
    $(getconf XBS5_LP64_OFF64_LDFLAGS) foo.c -o foo \
    $(getconf XBS5_LP64_OFF64_LIBS) -lxnet
```

同様の SUSv3 に準拠した LP64 アプリケーションを、次の形式のコマンド行で構築するようにしてください:

```
c99 $(getconf POSIX_V6_LP64_OFF64_CFLAGS) -D_XOPEN_SOURCE=600 \
    $(getconf POSIX_V6_LP64_OFF64_LDFLAGS) foo.c -o foo \
    $(getconf POSIX_V6_LP64_OFF64_LIBS) -lxnet
```

SUSv3

```
c99                         _XOPEN_SOURCE=600
```

関連項目

csh(1)、ksh(1)、sh(1)、exec(2)、sysconf(3C)、system(3C)、environ(5)、lf64(5)

名前

sticky – mark files for special treatment

機能説明

The *sticky bit* (file mode bit `01000`, see chmod(2)) is used to indicate special treatment of certain files and directories. A directory for which the sticky bit is set restricts deletion of files it contains. A file in a sticky directory can only be removed or renamed by a user who has write permission on the directory, and either owns the file, owns the directory, has write permission on the file, or is a privileged user. Setting the sticky bit is useful for directories such as `/tmp`, which must be publicly writable but should deny users permission to arbitrarily delete or rename the files of others.

If the sticky bit is set on a regular file and no execute bits are set, the system's page cache will not be used to hold the file's data. This bit is normally set on swap files of diskless clients so that accesses to these files do not flush more valuable data from the system's cache. Moreover, by default such files are treated as swap files, whose inode modification times may not necessarily be correctly recorded on permanent storage.

Any user may create a sticky directory. See `chmod` for details about modifying file modes.

関連項目

chmod(1), chmod(2), chown(2), mkdir(2), rename(2), unlink(2)

使用上の留意点

The mkdir(2) function will not create a directory with the sticky bit set.

名前 suri – storage URIs

機能説明 Storage URIs uniquely identify storage objects across different nodes.

The following Storage URIs are supported:

Dev URI

```
dev:<local-path-under-/dev>
dev:///<path-with-dev>
dev:<absolute-path-with-dev>
```

Examples:

```
dev:dsk/c0t0d0s0
dev:///dev/dsk/c0t0d0s0
dev:/dev/dsk/c0t0d0s0
```

The dev URI specifies an object in the form of an existing device. All forms of the dev URI are functionally equivalent. The "`dev:dsk`" format with no authority section and an implied "`/dev/`" prefix is usually preferred. The path must be a device in the `/dev` name space.

Components of the device path may only include characters from POSIX.1-2008 Portable Filename Character Set, 0-9a-zA-Z._-.

Logical Unit URI

```
lu:luname.naa.<ID>
lu:initiator.naa.<ID>,target.naa.<ID>,luname.naa.<ID>
```

Examples:

```
lu:luname.naa.5000c5000288fa25
lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.naa.5000c5000288fa25
```

Logical unit URI specifies a logical unit attached via fibre channel or serial-attached SCSI. In the luname-only URI form, the ID describes a logical unit name. In the initiator,target,luname form, an initiator specifies an initiator port and a target specifies a target port, and together they specify a path to the logical unit whose name must match the URI's luname part as in the luname-only form.

IDs are represented by using the SCSI-3 VPD page (0x83) global unique identifier (GUID) formatted as hexadecimal numbers (64/128 bits). IDs are case insensitive. Only GUIDs based on NAA identifiers are supported. See NOTES for information on NAA identifier format.

In general it is recommended using multipathing in conjunction with luname-only URIs. If multipathing is disabled and a luname-only URI is used, a random path to the specified logical unit will be chosen. To avoid this random path assignment an initiator,target,luname URI form can be used to select a specific path to a logical unit. If multipathing is enabled and an initiator,target,luname URI form is used then the multipathing framework controls which paths are used to access the logical unit and the URI is only used to identify the unit, not the access path.

If an initiator or/and a target IDs are not matched but the logical unit is found based on a logical unit name, the unit will be accessed via an alternate path.

iSCSI URI

```
iscsi:///luname.naa.<ID>
iscsi://<host>[:<port>]/luname.naa.<ID>
```

Examples:

```
iscsi:///luname.naa.600144f03d70c80000004ea57da10001
iscsi://[::1]/luname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1/luname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1:3620/luname.naa.600144f03d70c80000004ea57da10001
iscsi://hostname:3620/luname.naa.600144f03d70c80000004ea57da10001
```

The iSCSI URI specifies an object based on a logical unit number accessible via iSCSI protocol. Optional `hostname`[:`<port>`] authority section provides information for a consumer to set up a send-targets discovery address. IPv6 addresses must be enclosed in square brackets.

Format of an ID is explained in the “Logical Unit URI” section. See NOTES for information on NAA identifier format.

Getting URIs

To avoid errors by generating URIs manually, the `suriadm(1M)` subcommand `lookup-uri` is recommended to be used to generate URIs based on existing system device paths. Such device paths must be identified via other means. For Fibre Channel devices, `fcadm(1M)` may be used. For Serial Attached SCSI devices, `sasinfo(1M)` may be used. For iSCSI devices, `iscsiadm(1M)` may be used.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/library/storage/suri
Interface Stability	Commited

関連項目

`fcadm(1M)`, `iscsiadm(1M)`, `sasinfo(1M)`, `suriadm(1M)`, `attributes(5)`

Small Computer System Interface-3 (SCSI-3)

注意事項

NAA stands for “Name Address Authority” and it is a field within an identifier that specifies the format and length of that identifier. NAA identifier format is defined in SCSI Primary Commands - 3 specification, section “7.6.3.6 NAA identifier format”. WWN (World Wide Name) identifiers use an NAA identifier format.

名前

tecla, teclarc – User interface provided by the tecla library.

機能説明

This man page describes the command-line editing features that are available to users of programs that read keyboard input via the tecla library. Users of the `tcsh` `shell` will find the default key bindings very familiar. Users of the `bash` shell will also find it quite familiar, but with a few minor differences, most notably in how forward and backward searches through the list of historical commands are performed. There are two major editing modes, one with `emacs`-like key bindings and another with `vi`-like key bindings. By default `emacs` mode is enabled, but `vi(1)` mode can alternatively be selected via the user's configuration file. This file can also be used to change the bindings of individual keys to suit the user's preferences. By default, tab completion is provided. If the application hasn't reconfigured this to complete other types of symbols, then tab completion completes file names.

Key Sequence Notation

In the rest of this man page, and also in all tecla configuration files, key sequences are expressed as follows.

`^A` or `C-a`	This is a 'CONTROL-A', entered by pressing the CONTROL key at the same time as the 'A' key.
`\E` or `M-`	In key sequences, both of these notations can be entered either by pressing the ESCAPE key, then the following key, or by pressing the META key at the same time as the following key. Thus the key sequence `M-p` can be typed in two ways, by pressing the ESCAPE key, followed by pressing 'P', or by pressing the META key at the same time as 'P'.
up	This refers to the up-arrow key.
down	This refers to the down-arrow key.
left	This refers to the left-arrow key.
right	This refers to the right-arrow key.
a	This is just a normal 'A' key.

The Tecla Configuration File

By default, tecla looks for a file called `.teclarc` in your home directory (ie. `~/.teclarc`). If it finds this file, it reads it, interpreting each line as defining a new key binding or an editing configuration option. Since the `emacs` key-bindings are installed by default, if you want to use the non-default `vi` editing mode, the most important item to go in this file is the following line:

```
edit-mode vi
```

This will re-configure the default bindings for `vi`-mode. The complete set of arguments that this command accepts are:

vi	Install key bindings like those of the `vi` editor.
emacs	Install key bindings like those of the `emacs` editor. This is the default.
none	Use just the native line editing facilities provided by the terminal driver.

To prevent the terminal bell from being rung, such as when an unrecognized control-sequence is typed, place the following line in the configuration file:

```
nobeep
```

An example of a key binding line in the configuration file is the following.

```
bind M-[2~ insert-mode
```

On many keyboards, the above key sequence is generated when one presses the insert key, so with this key binding, one can toggle between the `emacs`-mode insert and overwrite modes by hitting one key. One could also do it by typing out the above sequence of characters one by one. As explained above, the `M-` part of this sequence can be typed either by pressing the ESCAPE key before the following key, or by pressing the META key at the same time as the following key. Thus if you had set the above key binding, and the insert key on your keyboard didn't generate the above key sequence, you could still type it in either of the following 2 ways.

1. Hit the ESCAPE key momentarily, then press '[', then '2', then finally '~'.
2. Press the META key at the same time as pressing the '[' key, then press '2', then '~'.

If you set a key binding for a key sequence that is already bound to a function, the new binding overrides the old one. If in the new binding you omit the name of the new function to bind to the key sequence, the original binding becomes undefined.

Starting with versions of `libtecla` later than 1.3.3 it is now possible to bind key sequences that begin with a printable character. Previously key sequences were required to start with a CONTROL or META character.

Note that the special keywords "up", "down", "left", and "right" refer to the arrow keys, and are thus not treated as key sequences. So, for example, to rebind the up and down arrow keys to use the history search mechanism instead of the simple history recall method, you could place the following in your configuration file:

```
bind up history-search-backwards
bind down history-search-backwards
```

To unbind an existing binding, you can do this with the bind command by omitting to name any action to rebind the key sequence to. For example, by not specifying an action function, the following command unbinds the default beginning-of-line action from the `^A` key sequence:

```
bind ^A
```

If you create a `~/.teclarc` configuration file, but it appears to have no effect on the program, check the documentation of the program to see if the author chose a different name for this file.

Filename and Tilde Completion

With the default key bindings, pressing the TAB key (aka. ^I) results in tecla attempting to complete the incomplete file name that precedes the cursor. Tecla searches backwards from the cursor, looking for the start of the file name, stopping when it hits either a space or the start of the line. If more than one file has the specified prefix, then tecla completes the file name up to the point at which the ambiguous matches start to differ, then lists the possible matches.

In addition to literally written file names, tecla can complete files that start with ~/ and ~user/ expressions and that contain $envvar expressions. In particular, if you hit TAB within an incomplete ~user, expression, tecla will attempt to complete the username, listing any ambiguous matches.

The completion binding is implemented using the cpl_complete_word() function, which is also available separately to users of this library. See the cpl_complete_word(3TECLA) man page for more details.

Filename Expansion

With the default key bindings, pressing ^X* causes tecla to expand the file name that precedes the cursor, replacing ~/ and ~user/ expressions with the corresponding home directories, and replacing $envvar expressions with the value of the specified environment variable, then if there are any wildcards, replacing the so far expanded file name with a space-separated list of the files which match the wild cards.

The expansion binding is implemented using the ef_expand_file() function. See the ef_expand_file(3TECLA) man page for more details.

Recalling Previously Typed Lines

Every time that a new line is entered by the user, it is appended to a list of historical input lines maintained within the GetLine resource object. You can traverse up and down this list using the up and down arrow keys. Alternatively, you can do the same with the ^P, and ^N keys, and in vi command mode you can alternatively use the k and j characters. Thus pressing up-arrow once, replaces the current input line with the previously entered line. Pressing up-arrow again, replaces this with the line that was entered before it, etc.. Having gone back one or more lines into the history list, one can return to newer lines by pressing down-arrow one or more times. If you do this sufficient times, you will return to the original line that you were entering when you first hit up-arrow.

Note that in vi mode, all of the history recall functions switch the library into command mode.

In emacs mode the M-p and M-n keys work just like the ^P and ^N keys, except that they skip all but those historical lines which share the prefix that precedes the cursor. In vi command mode the upper case 'K' and 'J' characters do the same thing, except that the string that they search for includes the character under the cursor as well as what precedes it.

Thus for example, suppose that you were in emacs mode, and you had just entered the following list of commands in the order shown:

```
ls ~/tecla/
cd ~/tecla
ls -l getline.c
emacs ~/tecla/getline.c
```

If you next typed:

```
ls
```

and then hit `M-p`, then rather than returning the previously typed `emacs` line, which doesn't start with "ls", tecla would recall the "ls -l getline.c" line. Pressing `M-p` again would recall the "ls ~/tecla/" line.

Note that if the string that you are searching for, contains any of the special characters, *, ?, or '[', then it is interpretted as a pattern to be matched. Thus, cotinuing with the above example, after typing in the list of commands shown, if you then typed:

```
*tecla*
```

and hit `M-p`, then the "`emacs` ~/tecla/getline.c" line would be recalled first, since it contains the word tecla somewhere in the line, Similarly, hitting `M-p` again, would recall the "ls ~/tecla/" line, and hitting it once more would recall the "ls ~/tecla/" line. The pattern syntax is the same as that described for file name expansion, in the `ef_expand_file`(3TECLA).

History Files

Authors of programs that use the tecla library have the option of saving historical command-lines in a file before exiting, and subsequently reading them back in from this file when the program is next started. There is no standard name for this file, since it makes sense for each application to use its own history file, so that commands from different applications don't get mixed up.

International Character Sets

Since `libtecla` version 1.4.0, tecla has been 8-bit clean. This means that all 8-bit characters that are printable in the user's current locale are now displayed verbatim and included in the returned input line. Assuming that the calling program correctly contains a call like the following,

```
setlocale(LC_CTYPE, "");
```

then the current locale is determined by the first of the environment variables `LC_CTYPE`, `LC_ALL`, and `LANG`, that is found to contain a valid locale name. If none of these variables are defined, or the program neglects to call `setlocale`, then the default C locale is used, which is US 7-bit ASCII. On most unix-like platforms, you can get a list of valid locales by typing the command:

```
locale -a
```

at the shell prompt.

Meta Keys and Locales

Beware that in most locales other than the default C locale, META characters become printable, and they are then no longer considered to match `M-c` style key bindings. This allows international characters to be entered with the compose key without unexpectedly triggering META key bindings. You can still invoke META bindings, since there are actually two ways to do this. For example the binding `M-c` can also be invoked by pressing the ESCAPE key momentarily, then pressing the c key, and this will work regardless of locale. Moreover, many modern terminal emulators, such as gnome's gnome-terminal's and KDE's konsole terminals, already generate escape pairs like this when you use the META key, rather than a real meta character, and other emulators usually have a way to request this behavior, so you can continue to use the META key on most systems.

For example, although xterm terminal emulators generate real 8-bit meta characters by default when you use the META key, they can be configured to output the equivalent escape pair by setting their `EightBitInput` X resource to False. You can either do this by placing a line like the following in your `~/.Xdefaults` file,

```
XTerm*EightBitInput: False
```

or by starting an `xterm` with an `-xrm` '*EightBitInput: False' command-line argument. In recent versions of xterm you can toggle this feature on and off with the 'Meta Sends Escape' option in the menu that is displayed when you press the left mouse button and the CONTROL key within an xterm window. In CDE, dtterms can be similarly coerced to generate escape pairs in place of meta characters, by setting the `Dtterm*KshMode` resource to True.

Entering International Characters

If you don't have a keyboard that generates all of the international characters that you need, there is usually a compose key that will allow you to enter special characters, or a way to create one. For example, under X windows on unix-like systems, if your keyboard doesn't have a compose key, you can designate a redundant key to serve this purpose with the xmodmap command. For example, on many PC keyboards there is a microsoft-windows key, which is otherwise useless under Linux. On a laptop, for example, the `xev` program might report that pressing this key generates keycode 115. To turn this key into a COMPOSE key, do the following:

```
xmodmap -e 'keycode 115 = Multi_key'
```

Type this key followed by a " character to enter an 'I' with a umlaut over it.

The Available Key Binding Functions

The following is a list of the editing functions provided by the tecla library. The names in the leftmost column of the list can be used in configuration files to specify which function a given key or combination of keys should invoke. They are also used in the next two sections to list the default key bindings in `emacs` and `vi` modes.

user-interrupt	Send a SIGINT signal to the parent process.
suspend	Suspend the parent process.

stop-output	Pause terminal output.
start-output	Resume paused terminal output.
literal-next	Arrange for the next character to be treated as a normal character. This allows control characters to be entered.
cursor-right	Move the cursor one character right.
cursor-left	Move the cursor one character left.
insert-mode	Toggle between insert mode and overwrite mode.
beginning-of-line	Move the cursor to the beginning of the line.
end-of-line	Move the cursor to the end of the line.
delete-line	Delete the contents of the current line.
kill-line	Delete everything that follows the cursor.
backward-kill-line	Delete all characters between the cursor and the start of the line.
forward-word	Move to the end of the word which follows the cursor.
forward-to-word	Move the cursor to the start of the word that follows the cursor.
backward-word	Move to the start of the word which precedes the cursor.
goto-column	Move the cursor to the 1-relative column in the line specified by any preceding digit-argument sequences (see Entering Repeat Counts below).
find-parenthesis	If the cursor is currently over a parenthesis character, move it to the matching parenthesis character. If not over a parenthesis character move right to the next close parenthesis.
forward-delete-char	Delete the character under the cursor.
backward-delete-char	Delete the character which precedes the cursor.
list-or-eof	This is intended for binding to `^D`. When invoked when the cursor is within the line it displays all possible completions then redisplays the line unchanged. When invoked on an empty line, it signals end-of-input (EOF) to the caller of `gl_get_line()`.
del-char-or-list-or-eof	This is intended for binding to `^D`. When invoked when the cursor is within the line it invokes forward-delete-char. When invoked at the end of the line it displays all possible

	completions then redisplays the line unchanged. When invoked on an empty line, it signals end-of-input (EOF) to the caller of `gl_get_line()`.
forward-delete-word	Delete the word which follows the cursor.
backward-delete-word	Delete the word which precedes the cursor.
upcase-word	Convert all of the characters of the word which follows the cursor, to upper case.
downcase-word	Convert all of the characters of the word which follows the cursor, to lower case.
capitalize-word	Capitalize the word which follows the cursor.
change-case	If the next character is upper case, toggle it to lower case and vice versa.
redisplay	Redisplay the line.
clear-screen	Clear the terminal, then redisplay the current line.
transpose-chars	Swap the character under the cursor with the character just before the cursor.
set-mark	Set a mark at the position of the cursor.
exchange-point-and-mark	Move the cursor to the last mark that was set, and move the mark to where the cursor used to be.
kill-region	Delete the characters that lie between the last mark that was set, and the cursor.
copy-region-as-kill	Copy the text between the mark and the cursor to the cut buffer, without deleting the original text.
yank	Insert the text that was last deleted, just before the current position of the cursor.
append-yank	Paste the current contents of the cut buffer, after the cursor.
up-history	Recall the next oldest line that was entered. Note that in `vi` mode you are left in command mode.
down-history	Recall the next most recent line that was entered. If no history recall session is currently active, the next line from a previous recall session is recalled. Note that in vi mode you are left in command mode.
history-search-backward	Recall the next oldest line who's prefix matches the string which currently precedes the cursor (in `vi` command-mode

	the character under the cursor is also included in the search string). Note that in `vi` mode you are left in command mode.
history-search-forward	Recall the next newest line who's prefix matches the string which currently precedes the cursor (in `vi` command-mode the character under the cursor is also included in the search string). Note that in `vi` mode you are left in command mode.
history-re-search-backward	Recall the next oldest line who's prefix matches that established by the last invocation of either history-search-forward or history-search-backward.
history-re-search-forward	Recall the next newest line who's prefix matches that established by the last invocation of either history-search-forward or history-search-backward.
complete-word	Attempt to complete the incomplete word which precedes the cursor. Unless the host program has customized word completion, file name completion is attempted. In `vi` commmand mode the character under the cursor is also included in the word being completed, and you are left in `vi` insert mode.
expand-filename	Within the command line, expand wild cards, tilde expressions and dollar expressions in the file name which immediately precedes the cursor. In `vi` commmand mode the character under the cursor is also included in the file name being expanded, and you are left in `vi` insert mode.
list-glob	List any file names which match the wild-card, tilde and dollar expressions in the file name which immediately precedes the cursor, then redraw the input line unchanged.
list-history	Display the contents of the history list for the current history group. If a repeat count of > `1` is specified, only that many of the most recent lines are displayed. See the Entering Repeat Counts section.
read-from-file	Temporarily switch to reading input from the file who's name precedes the cursor.
read-init-files	Re-read `teclarc` configuration files.
beginning-of-history	Move to the oldest line in the history list. Note that in `vi` mode you are left in command mode.
end-of-history	Move to the newest line in the history list (ie. the current line). Note that in `vi` mode this leaves you in command mode.

digit-argument	Enter a repeat count for the next key binding function. For details, see the Entering Repeat Counts section.
newline	Terminate and return the current contents of the line, after appending a newline character. The newline character is normally '\n', but will be the first character of the key sequence that invoked the newline action, if this happens to be a printable character. If the action was invoked by the '\n' newline character or the '\r' carriage return character, the line is appended to the history buffer.
repeat-history	Return the line that is being edited, then arrange for the next most recent entry in the history buffer to be recalled when tecla is next called. Repeatedly invoking this action causes successive historical input lines to be re-executed. Note that this action is equivalent to the 'Operate' action in ksh.
ring-bell	Ring the terminal bell, unless the bell has been silenced via the nobeep configuration option (see The Tecla Configuration File section).
forward-copy-char	Copy the next character into the cut buffer (NB. use repeat counts to copy more than one).
backward-copy-char	Copy the previous character into the cut buffer.
forward-copy-word	Copy the next word into the cut buffer.
backward-copy-word	Copy the previous word into the cut buffer.
forward-find-char	Move the cursor to the next occurrence of the next character that you type.
backward-find-char	Move the cursor to the last occurrence of the next character that you type.
forward-to-char	Move the cursor to the character just before the next occurrence of the next character that the user types.
backward-to-char	Move the cursor to the character just after the last occurrence before the cursor of the next character that the user types.
repeat-find-char	Repeat the last backward-find-char, forward-find-char, backward-to-char or forward-to-char.
invert-refind-char	Repeat the last backward-find-char, forward-find-char, backward-to-char, or forward-to-char in the opposite direction.

delete-to-column	Delete the characters from the cursor up to the column that is specified by the repeat count.
delete-to-parenthesis	Delete the characters from the cursor up to and including the matching parenthesis, or next close parenthesis.
forward-delete-find	Delete the characters from the cursor up to and including the following occurence of the next character typed.
backward-delete-find	Delete the characters from the cursor up to and including the preceding occurence of the next character typed.
forward-delete-to	Delete the characters from the cursor up to, but not including, the following occurence of the next character typed.
backward-delete-to	Delete the characters from the cursor up to, but not including, the preceding occurence of the next character typed.
delete-refind	Repeat the last *-delete-find or *-delete-to action.
delete-invert-refind	Repeat the last *-delete-find or *-delete-to action, in the opposite direction.
copy-to-column	Copy the characters from the cursor up to the column that is specified by the repeat count, into the cut buffer.
copy-to-parenthesis	Copy the characters from the cursor up to and including the matching parenthesis, or next close parenthesis, into the cut buffer.
forward-copy-find	Copy the characters from the cursor up to and including the following occurence of the next character typed, into the cut buffer.
backward-copy-find	Copy the characters from the cursor up to and including the preceding occurence of the next character typed, into the cut buffer.
forward-copy-to	Copy the characters from the cursor up to, but not including, the following occurence of the next character typed, into the cut buffer.
backward-copy-to	Copy the characters from the cursor up to, but not including, the preceding occurence of the next character typed, into the cut buffer.
copy-refind	Repeat the last *-copy-find or *-copy-to action.

copy-invert-refind	Repeat the last *-copy-find or *-copy-to action, in the opposite direction.
vi-mode	Switch to `vi` mode from emacs mode.
emacs-mode	Switch to `emacs` mode from `vi` mode.
vi-insert	From `vi` command mode, switch to insert mode.
vi-overwrite	From `vi` command mode, switch to overwrite mode.
vi-insert-at-bol	From `vi` command mode, move the cursor to the start of the line and switch to insert mode.
vi-append-at-eol	From `vi` command mode, move the cursor to the end of the line and switch to append mode.
vi-append	From `vi` command mode, move the cursor one position right, and switch to insert mode.
vi-replace-char	From `vi` command mode, replace the character under the cursor with the next character entered.
vi-forward-change-char	From `vi` command mode, delete the next character then enter insert mode.
vi-backward-change-char	From vi command mode, delete the preceding character then enter insert mode.
vi-forward-change-word	From `vi` command mode, delete the next word then enter insert mode.
vi-backward-change-word	From vi command mode, delete the preceding word then enter insert mode.
vi-change-rest-of-line	From `vi` command mode, delete from the cursor to the end of the line, then enter insert mode.
vi-change-line	From `vi` command mode, delete the current line, then enter insert mode.
vi-change-to-bol	From `vi` command mode, delete all characters between the cursor and the beginning of the line, then enter insert mode.
vi-change-to-column	From `vi` command mode, delete the characters from the cursor up to the column that is specified by the repeat count, then enter insert mode.
vi-change-to-parenthesis	Delete the characters from the cursor up to and including the matching parenthesis, or next close parenthesis, then enter `vi` insert mode.

vi-forward-change-find	From `vi` command mode, delete the characters from the cursor up to and including the following occurence of the next character typed, then enter insert mode.
vi-backward-change-find	From vi command mode, delete the characters from the cursor up to and including the preceding occurence of the next character typed, then enter insert mode.
vi-forward-change-to	From `vi` command mode, delete the characters from the cursor up to, but not including, the following occurence of the next character typed, then enter insert mode.
vi-backward-change-to	From `vi` command mode, delete the characters from the cursor up to, but not including, the preceding occurence of the next character typed, then enter insert mode.
vi-change-refind	Repeat the last vi-*-change-find or vi-*-change-to action.
vi-change-invert-refind	Repeat the last vi-*-change-find or vi-*-change-to action, in the opposite direction.
vi-undo	In `vi` mode, undo the last editing operation.
vi-repeat-change	In `vi` command mode, repeat the last command that modified the line.

Default Key Bindings In emacs Mode

The following default key bindings, which can be overriden by the tecla configuration file, are designed to mimic most of the bindings of the unix `tcsh` `shell` shell, when it is in `emacs` editing mode.

This is the default editing mode of the tecla library.

Under UNIX the terminal driver sets a number of special keys for certain functions. The tecla library attempts to use the same key bindings to maintain consistency. The key sequences shown for the following 6 bindings are thus just examples of what they will probably be set to. If you have used the stty command to change these keys, then the default bindings should match.

`^C`	user-interrupt
`^\literal>`	abort
`^Z`	suspend
`^Q`	start-output
`^S`	stop-output
`^V`	literal-next

The cursor keys are refered to by name, as follows. This is necessary because different types of terminals generate different key sequences when their cursor keys are pressed.

right	cursor-right
left	cursor-left
up	up-history
down	down-history

The remaining bindings don't depend on the terminal setttings.

`^F`	cursor-right
`^B`	cursor-left
`M-i`	insert-mode
`^A`	beginning-of-line
`^E`	end-of-line
`^U`	delete-line
`^K`	kill-line
`M-f`	forward-word
`M-b`	backward-word
`^D`	del-char-or-list-or-eof
`^H`	backward-delete-char
`^?`	backward-delete-char
`M-d`	forward-delete-word
`M-^H`	backward-delete-word
`M-^?`	backward-delete-word
`M-u`	upcase-word
`M-l`	downcase-word
`M-c`	capitalize-word
`^R`	redisplay
`^L`	clear-screen
`^T`	transpose-chars
`^@`	set-mark

^X^X	exchange-point-and-mark
^W	kill-region
M-w	copy-region-as-kill
^Y	yank
^P	up-history
^N	down-history
M-p	history-search-backward
M-n	history-search-forward
^I	complete-word
^X*	expand-filename
^X^F	read-from-file
^X^R	read-init-files
^Xg	list-glob
^Xh	list-history
M-<	beginning-of-history
M->	end-of-history
\n	newline
\r	newline
M-o	repeat-history
M-^V	vi-mode
M-0, M-1, ... M-9	digit-argument (see below)

Note that ^I is what the TAB key generates, and that ^@ can be generated not only by pressing the CONTROL key and the @ key simultaneously, but also by pressing the CONTROL key and the space bar at the same time.

Default Key Bindings in vi Mode

The following default key bindings are designed to mimic the vi style of editing as closely as possible. This means that very few editing functions are provided in the initial character input mode, editing functions instead being provided by the vi command mode. The vi command mode is entered whenever the ESCAPE character is pressed, or whenever a key sequence that starts with a meta character is entered. In addition to mimicing vi, libtecla provides bindings for tab completion, wild-card expansion of file names, and historical line recall.

To learn how to tell the tecla library to use `vi` mode instead of the default `emacs` editing mode, see the earlier section entitled The Tecla Configuration File.

Under UNIX the terminal driver sets a number of special keys for certain functions. The tecla library attempts to use the same key bindings to maintain consistency, binding them both in input mode and in command mode. The key sequences shown for the following 6 bindings are thus just examples of what they will probably be set to. If you have used the `stty` command to change these keys, then the default bindings should match.

`^C`	user-interrupt
`^\`	abort
`^Z`	suspend
`^Q`	start-output
`^S`	stop-output
`^V`	literal-next
`M-^C`	user-interrupt
`M-^\literal>`	abort
`M-^Z`	suspend
`M-^Q`	start-output
`M-^S`	stop-output

Note that above, most of the bindings are defined twice, once as a raw control code like `^C` and then a second time as a META character like `M-^C`. The former is the binding for `vi` input mode, whereas the latter is the binding for `vi` command mode. Once in command mode all key sequences that the user types that they don't explicitly start with an ESCAPE or a META key, have their first key secretly converted to a META character before the key sequence is looked up in the key binding table. Thus, once in command mode, when you type the letter i, for example, the tecla library actually looks up the binding for `M-i`.

The cursor keys are refered to by name, as follows. This is necessary because different types of terminals generate different key sequences when their cursor keys are pressed.

`right`	cursor-right
`left`	cursor-left
`up`	up-history
`down`	down-history

The cursor keys normally generate a key sequence that start with an ESCAPE character, so beware that using the arrow keys will put you into command mode (if you aren't already in command mode).

The following are the terminal-independent key bindings for `vi` input mode.

`^D`	list-or-eof
`^G`	list-glob
`^H`	backward-delete-char
`^I`	complete-word
`\r`	newline
`\n`	newline
`^L`	clear-screen
`^N`	down-history
`^P`	up-history
`^R`	redisplay
`^U`	backward-kill-line
`^W`	backward-delete-word
`^X*`	expand-filename
`^X^F`	read-from-file
`^X^R`	read-init-files
`^?`	backward-delete-char

The following are the key bindings that are defined in `vi` command mode, this being specified by them all starting with a META character. As mentioned above, once in command mode the initial meta character is optional. For example, you might enter command mode by typing ESCAPE, and then press 'H' twice to move the cursor two positions to the left. Both 'H' characters get quietly converted to `M-h` before being compared to the key binding table, the first one because ESCAPE followed by a character is always converted to the equivalent META character, and the second because command mode was already active.

`M-\`	cursor-right (META-space)
`M-$`	end-of-line
`M-*`	expand-filename
`M-+`	down-history

M--	up-history
M-<	beginning-of-history
M->	end-of-history
M-^	beginning-of-line
M-	repeat-find-char
M-,	invert-refind-char
M-|	goto-column
M-~	change-case
M-.	vi-repeat-change
M-%	find-parenthesis
M-a	vi-append
M-A	vi-append-at-eol
M-b	backward-word
M-B	backward-word
M-C	vi-change-rest-of-line
M-cb	vi-backward-change-word
M-cB	vi-backward-change-word
M-cc	vi-change-line
M-ce	vi-forward-change-word
M-cE	vi-forward-change-word
M-cw	vi-forward-change-word
M-cW	vi-forward-change-word
M-cF	vi-backward-change-find
M-cf	vi-forward-change-find
M-cT	vi-backward-change-to
M-ct	vi-forward-change-to
M-c;	vi-change-refind
M-c,	vi-change-invert-refind
M-ch	vi-backward-change-char

M-c^H	vi-backward-change-char
M-c^?	vi-backward-change-char
M-cl	vi-forward-change-char
M-c\	vi-forward-change-char (META-c-space)
M-c^	vi-change-to-bol
M-c0	vi-change-to-bol
M-c$	vi-change-rest-of-line
M-c|	vi-change-to-column
M-c%	vi-change-to-parenthesis
M-dh	backward-delete-char
M-d^H	backward-delete-char
M-d^?	backward-delete-char
M-dl	forward-delete-char
M-d	forward-delete-char (META-d-space)
M-dd	delete-line
M-db	backward-delete-word
M-dB	backward-delete-word
M-de	forward-delete-word
M-dE	forward-delete-word
M-dw	forward-delete-word
M-dW	forward-delete-word
M-dF	backward-delete-find
M-df	forward-delete-find
M-dT	backward-delete-to
M-dt	forward-delete-to
M-d;	delete-refind
M-d,	delete-invert-refind
M-d^	backward-kill-line
M-d0	backward-kill-line

M-d$	kill-line
M-D	kill-line
M-d|	delete-to-column
M-d%	delete-to-parenthesis
M-e	forward-word
M-E	forward-word
M-f	forward-find-char
M-F	backward-find-char
M--	up-history
M-h	cursor-left
M-H	beginning-of-history
M-i	vi-insert
M-I	vi-insert-at-bol
M-j	down-history
M-J	history-search-forward
M-k	up-history
M-K	history-search-backward
M-l	cursor-right
M-L	end-of-history
M-n	history-re-search-forward
M-N	history-re-search-backward
M-p	append-yank
M-P	yank
M-r	vi-replace-char
M-R	vi-overwrite
M-s	vi-forward-change-char
M-S	vi-change-line
M-t	forward-to-char
M-T	backward-to-char

M-u	vi-undo
M-w	forward-to-word
M-W	forward-to-word
M-x	forward-delete-char
M-X	backward-delete-char
M-yh	backward-copy-char
M-y^H	backward-copy-char
M-y^?	backward-copy-char
M-yl	forward-copy-char
M-y\	forward-copy-char (META-y-space)
M-ye	forward-copy-word
M-yE	forward-copy-word
M-yw	forward-copy-word
M-yW	forward-copy-word
M-yb	backward-copy-word
M-yB	backward-copy-word
M-yf	forward-copy-find
M-yF	backward-copy-find
M-yt	forward-copy-to
M-yT	backward-copy-to
M-y;	copy-refind
M-y,	copy-invert-refind
M-y^	copy-to-bol
M-y0	copy-to-bol
M-y$	copy-rest-of-line
M-yy	copy-line
M-Y	copy-line
M-y|	copy-to-column
M-y%	copy-to-parenthesis

`M-^E`	emacs-mode
`M-^H`	cursor-left
`M-^?`	cursor-left
`M-^L`	clear-screen
`M-^N`	down-history
`M-^P`	up-history
`M-^R`	redisplay
`M-^D`	list-or-eof
`M-^I`	complete-word
`M-\r`	newline
`M-\n`	newline
`M-^X^R`	read-init-files
`M-^Xh`	list-history
`M-0, M-1, ... M-9`	digit-argument (see below)

Note that `^I` is what the TAB key generates.

Entering Repeat Counts

Many of the key binding functions described previously, take an optional count, typed in before the target key sequence. This is interpreted as a repeat count by most bindings. A notable exception is the goto-column binding, which interprets the count as a column number.

By default you can specify this count argument by pressing the META key while typing in the numeric count. This relies on the digit-argument action being bound to 'META-0', 'META-1' etc. Once any one of these bindings has been activated, you can optionally take your finger off the META key to type in the rest of the number, since every numeric digit thereafter is treated as part of the number, unless it is preceded by the literal-next binding. As soon as a non-digit, or literal digit key is pressed the repeat count is terminated and either causes the just typed character to be added to the line that many times, or causes the next key binding function to be given that argument.

For example, in `emacs` mode, typing:

```
M-12a
```

causes the letter 'a' to be added to the line 12 times, whereas

```
M-4M-c
```

Capitalizes the next 4 words.

In vi command mode the meta modifier is automatically added to all characters typed in, so to enter a count in vi command-mode, just involves typing in the number, just as it does in the vi editor itself. So for example, in vi command mode, typing:

```
4w2x
```

moves the cursor four words to the right, then deletes two characters.

You can also bind digit-argument to other key sequences. If these end in a numeric digit, that digit gets appended to the current repeat count. If it doesn't end in a numeric digit, a new repeat count is started with a value of zero, and can be completed by typing in the number, after letting go of the key which triggered the digit-argument action.

ファイル

`/usr/lib/libtecla.so`	The tecla library
`/usr/include/libtecla.h`	The tecla header file
`~/.teclarc`	The personal tecla customization file

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	library/libtecla
Interface Stability	Committed

関連項目

vi(1), cpl_complete_word(3TECLA), ef_expand_file(3TECLA), gl_get_line(3TECLA), gl_io_mode(3TECLA), libtecla(3LIB), pca_lookup_file(3TECLA), attributes(5)

名前

term – conventional names for terminals

機能説明

Terminal names are maintained as part of the shell environment in the environment variable `TERM`. See sh(1), profile(4), and environ(5). These names are used by certain commands (for example, `tabs`, `tput`, and `vi`) and certain functions (for example, see curses(3CURSES)).

Files under `/usr/share/lib/terminfo` are used to name terminals and describe their capabilities. These files are in the format described in terminfo(4). Entries in `terminfo` source files consist of a number of comma-separated fields. To print a description of a terminal *term*, use the command `infocmp -I` *term*. See infocmp(1M). White space after each comma is ignored. The first line of each terminal description in the `terminfo` database gives the names by which `terminfo` knows the terminal, separated by bar (`|`) characters. The first name given is the most common abbreviation for the terminal (this is the one to use to set the environment variable `TERMINFO` in `$HOME/.profile`; see profile(4)), the last name given should be a long name fully identifying the terminal, and all others are understood as synonyms for the terminal name. All names but the last should contain no blanks and must be unique in the first 14 characters; the last name may contain blanks for readability.

Terminal names (except for the last, verbose entry) should be chosen using the following conventions. The particular piece of hardware making up the terminal should have a root name chosen, for example, for the AT&T 4425 terminal, `att4425`. This name should not contain hyphens, except that synonyms may be chosen that do not conflict with other names. Up to 8 characters, chosen from the set `a` through `z` and `0` through `9`, make up a basic terminal name. Names should generally be based on original vendors rather than local distributors. A terminal acquired from one vendor should not have more than one distinct basic name. Terminal sub-models, operational modes that the hardware can be in, or user preferences should be indicated by appending a hyphen and an indicator of the mode. Thus, an AT&T 4425 terminal in 132 column mode is `att4425–w`. The following suffixes should be used where possible:

Suffix	Meaning	Example
–w	Wide mode (more than 80 columns)	att4425–w
–am	With auto. margins (usually default)	vt100–am
–nam	Without automatic margins	vt100–nam
–*n*	Number of lines on the screen	aaa–60
–na	No arrow keys (leave them in local)	c100–na
–np	Number of pages of memory	c100–4p
–rv	Reverse video	att4415–rv

To avoid conflicts with the naming conventions used in describing the different modes of a terminal (for example, `-w`), it is recommended that a terminal's root name not contain hyphens. Further, it is good practice to make all terminal names used in the `terminfo(4)` database unique. Terminal entries that are present only for inclusion in other entries via the `use=` facilities should have a '+' in their name, as in `4415+nl`.

Here are some of the known terminal names: (For a complete list, enter the command `ls -C /usr/share/lib/terminfo/?`).

2621,hp2621	Hewlett-Packard 2621 series
2631	Hewlett-Packard 2631 line printer
2631–c	Hewlett-Packard 2631 line printer, compressed mode
2631–e	Hewlett-Packard 2631 line printer, expanded mode
2640,hp2640	Hewlett-Packard 2640 series
2645,hp2645	Hewlett-Packard 2645 series
3270	IBM Model 3270
33,tty33	AT&T Teletype Model 33 KSR
35,tty35	AT&T Teletype Model 35 KSR
37,tty37	AT&T Teletype Model 37 KSR
4000a	Trendata 4000a
4014,tek4014	TEKTRONIX 4014
40,tty40	AT&T Teletype Dataspeed 40/2
43,tty43	AT&T Teletype Model 43 KSR
4410,5410	AT&T 4410/5410 in 80-column mode, version 2
4410–nfk,5410–nfk	AT&T 4410/5410 without function keys, version 1
4410–nsl,5410–nsl	AT&T 4410/5410 without pln defined
4410–w,5410–w	AT&T 4410/5410 in 132-column mode
4410v1,5410v1	AT&T 4410/5410 in 80-column mode, version 1
4410v1–w,5410v1–w	AT&T 4410/5410 in 132-column mode, version 1
4415,5420	AT&T 4415/5420 in 80-column mode
4415–nl,5420–nl	AT&T 4415/5420 without changing labels
4415–rv,5420–rv	AT&T 4415/5420 80 columns in reverse video

4415–rv–nl,5420–rv–nl	AT&T 4415/5420 reverse video without changing labels
4415–w,5420–w	AT&T 4415/5420 in 132-column mode
4415–w–nl,5420–w–nl	AT&T 4415/5420 in 132-column mode without changing labels
4415–w–rv,5420–w–rv	AT&T 4415/5420 132 columns in reverse video
4418,5418	AT&T 5418 in 80-column mode
4418–w,5418–w	AT&T 5418 in 132-column mode
4420	AT&T Teletype Model 4420
4424	AT&T Teletype Model 4424
4424-2	AT&T Teletype Model 4424 in display function group ii
4425,5425	AT&T 4425/5425
4425–fk,5425–fk	AT&T 4425/5425 without function keys
4425–nl,5425–nl	AT&T 4425/5425 without changing labels in 80-column mode
4425–w,5425–w	AT&T 4425/5425 in 132-column mode
4425–w–fk,5425–w–fk	AT&T 4425/5425 without function keys in 132-column mode
4425–nl–w,5425–nl–w	AT&T 4425/5425 without changing labels in 132-column mode
4426	AT&T Teletype Model 4426S
450	DASI 450 (same as Diablo 1620)
450–12	DASI 450 in 12-pitch mode
500,att500	AT&T-IS 500 terminal
510,510a	AT&T 510/510a in 80-column mode
513bct,att513	AT&T 513 bct terminal
5320	AT&T 5320 hardcopy terminal
5420_2	AT&T 5420 model 2 in 80-column mode
5420_2–w	AT&T 5420 model 2 in 132-column mode
5620,dmd	AT&T 5620 terminal 88 columns
5620–24,dmd–24	AT&T Teletype Model DMD 5620 in a 24x80 layer
5620–34,dmd–34	AT&T Teletype Model DMD 5620 in a 34x80 layer
610,610bct	AT&T 610 bct terminal in 80-column mode
610–w,610bct–w	AT&T 610 bct terminal in 132-column mode

630,630MTG	AT&T 630 Multi-Tasking Graphics terminal
7300,pc7300,unix_pc	AT&T UNIX PC Model 7300
735,ti	Texas Instruments TI735 and TI725
745	Texas Instruments TI745
dumb	generic name for terminals that lack reverse line-feed and other special escape sequences
hp	Hewlett-Packard (same as 2645)
lp	generic name for a line printer
pt505	AT&T Personal Terminal 505 (22 lines)
pt505–24	AT&T Personal Terminal 505 (24-line mode)
sync	generic name for synchronous Teletype Model 4540-compatible terminals

Commands whose behavior depends on the type of terminal should accept arguments of the form `-T`*term* where *term* is one of the names given above; if no such argument is present, such commands should obtain the terminal type from the environment variable `TERM`, which, in turn, should contain *term*.

ファイル

`/usr/share/lib/terminfo/?/*` compiled terminal description database

関連項目

`sh`(1), `stty`(1), `tabs`(1), `tput`(1), `vi`(1), `infocmp`(1M), `curses`(3CURSES), `profile`(4), `terminfo`(4), `environ`(5)

名前

threads, pthreads – POSIX pthread と Solaris スレッドの概念

形式

POSIX

```
cc –mt [ flag... ] file... [ -lrt library... ]
#include <pthread.h>
```

Solaris

```
cc –mt [ flag... ] file... [ library... ]
#include <sched.h>
#include <thread.h>
```

機能説明

POSIX スレッドと Solaris スレッドにはそれぞれ、libc(3LIB) 内に独自の実装があります。両方の実装は相互運用可能であり、それぞれの機能は類似しているため、同じアプリケーション内で使用できます。POSIX に準拠したほかの環境に完全に移植できることが保証されているのは POSIX スレッドだけです。POSIX スレッドと Solaris スレッドには、異なるソース、インクルードファイル、およびリンクライブラリが必要です。「形式」を参照してください。

類似点

POSIX と Solaris のスレッド化関数のほとんどには、互いに対応する関数があります。セマフォー名を除き、POSIX 関数名には「pthread」の接頭辞が付いています。類似した POSIX 関数と Solaris 関数は、関数名の末尾が類似しています。通常、類似した POSIX 関数と Solaris 関数には、同じ数と用途の引数があります。

相違点

POSIX pthread と Solaris スレッドは、次の点で異なっています:

- POSIX スレッドの方が移植性があります。
- POSIX スレッドでは、構成可能な属性オブジェクトに従って、各スレッドの特性が確立されています。
- POSIX pthread には、スレッド取り消しが実装されています。
- POSIX pthread では、スケジューリングのアルゴリズムが適用されます。
- POSIX pthread では、fork(2) 呼び出しのためのクリーンアップハンドラが許可されます。
- Solaris スレッドは中断したり、続行したりできます。
- Solaris スレッドには、プロセスがその消滅を待たないデーモンスレッドが実装されています。

関数の比較

次の表に、POSIX pthread と Solaris スレッドの関数の比較を示します。POSIX pthread または Solaris スレッドのどちらかで比較可能なインタフェースがない場合は、その列にハイフン (–) が表示されます。

作成に関連した関数

POSIX	Solaris

pthread_create()	thr_create()
pthread_attr_init()	–
pthread_attr_setdetachstate()	–
pthread_attr_getdetachstate()	–
pthread_attr_setinheritsched()	–
pthread_attr_getinheritsched()	–
pthread_attr_setschedparam()	–
pthread_attr_getschedparam()	–
pthread_attr_setschedpolicy()	–
pthread_attr_getschedpolicy()	–
pthread_attr_setscope()	–
pthread_attr_getscope()	–
pthread_attr_setstackaddr()	–
pthread_attr_getstackaddr()	–
pthread_attr_setstacksize()	–
pthread_attr_getstacksize()	–
pthread_attr_getguardsize()	–
pthread_attr_setguardsize()	–
pthread_attr_destroy()	–
–	thr_min_stack()

終了に関連した関数

POSIX	Solaris
pthread_exit()	thr_exit()
pthread_join()	thr_join()
pthread_detach()	–

スレッド固有のデータに関連した関数

POSIX	Solaris
pthread_key_create()	thr_keycreate()
pthread_setspecific()	thr_setspecific()

pthread_getspecific()	thr_getspecific()
pthread_key_delete()	–

シグナルに関連した関数

POSIX	Solaris
pthread_sigmask()	thr_sigsetmask()
pthread_kill()	thr_kill()

ID に関連した関数

POSIX	Solaris
pthread_self()	thr_self()
pthread_equal()	–
–	thr_main()

スケジューリングに関連した関数

POSIX	Solaris
–	thr_yield()
–	thr_suspend()
–	thr_continue()
pthread_setconcurrency()	thr_setconcurrency()
pthread_getconcurrency()	thr_getconcurrency()
pthread_setschedparam()	thr_setprio()
pthread_setschedprio()	thr_setprio()
pthread_getschedparam()	thr_getprio()

取り消しに関連した関数

POSIX	Solaris
pthread_cancel()	–
pthread_setcancelstate()	–
pthread_setcanceltype()	–
pthread_testcancel()	–
pthread_cleanup_pop()	–
pthread_cleanup_push()	–

相互排他ロックに関連した関数

POSIX	Solaris
pthread_mutex_init()	mutex_init()
pthread_mutexattr_init()	—
pthread_mutexattr_setpshared()	—
pthread_mutexattr_getpshared()	—
pthread_mutexattr_setprotocol()	—
pthread_mutexattr_getprotocol()	—
pthread_mutexattr_setprioceiling()	—
pthread_mutexattr_getprioceiling()	—
pthread_mutexattr_settype()	—
pthread_mutexattr_gettype()	—
pthread_mutexattr_setrobust()	—
pthread_mutexattr_getrobust()	—
pthread_mutexattr_destroy()	—
pthread_mutex_setprioceiling()	—
pthread_mutex_getprioceiling()	—
pthread_mutex_lock()	mutex_lock()
pthread_mutex_trylock()	mutex_trylock()
pthread_mutex_unlock()	mutex_unlock()
pthread_mutex_destroy()	mutex_destroy()

条件変数に関連した関数

POSIX	Solaris
pthread_cond_init()	cond_init()
pthread_condattr_init()	—
pthread_condattr_setpshared()	—
pthread_condattr_getpshared()	—
pthread_condattr_destroy()	—
pthread_cond_wait()	cond_wait()
pthread_cond_timedwait()	cond_timedwait()

pthread_cond_signal()	cond_signal()
pthread_cond_broadcast()	cond_broadcast()
pthread_cond_destroy()	cond_destroy()

読み取り書き込みロックに関連した関数

POSIX	Solaris
pthread_rwlock_init()	rwlock_init()
pthread_rwlock_rdlock()	rw_rdlock()
pthread_rwlock_tryrdlock()	rw_tryrdlock()
pthread_rwlock_wrlock()	rw_wrlock()
pthread_rwlock_trywrlock()	rw_trywrlock()
pthread_rwlock_unlock()	rw_unlock()
pthread_rwlock_destroy()	rwlock_destroy()
pthread_rwlockattr_init()	–
pthread_rwlockattr_destroy()	–
pthread_rwlockattr_getpshared()	–
pthread_rwlockattr_setpshared()	–

セマフォーに関連した関数

POSIX	Solaris
sem_init()	sema_init()
sem_open()	–
sem_close()	–
sem_wait()	sema_wait()
sem_trywait()	sema_trywait()
sem_post()	sema_post()
sem_getvalue()	–
sem_unlink()	–
sem_destroy()	sema_destroy()

fork( ) クリーンアップに関連した関数

POSIX	Solaris

pthread_atfork()	–

制限に関連した関数

POSIX	Solaris
pthread_once()	–

デバッグに関連した関数

POSIX	Solaris
–	thr_stksegment()

ロック

同期

マルチスレッド動作は非同期であるため、並列処理や並行処理のために最適化されます。各スレッドは (常に同じプロセス内から、場合によっては複数のプロセスから) 互いにグローバルデータを共有しますが、共有データへの排他的アクセスが任意の特定の時点で保証されているとは限りません。共有データへの相互に排他的なアクセスをセキュリティーで保護するには、スレッド間の同期が必要です。POSIX と Solaris のどちらにも、次の 4 つの同期メカニズムが実装されています: 相互排他ロック、条件変数、読み取り書き込みロック (頻繁な読み取りと不定期の書き込みのための最適化された相互排他ロック)、セマフォー。

複数のスレッドを同期させると、それらの並行性が低下します。同期の粒度が粗い程 (つまり、ロックされるコードのブロックが大きい程)、並行性は低くなります。

MT の fork()

スレッドのプログラムが fork(2) を呼び出した場合、呼び出し元スレッドのみを複製する fork1(2) が暗黙的に呼び出されます。プロセス全体にわたって未処理の相互排他ロックが存在する場合、アプリケーションは fork() を呼び出す前に、待機してこれらの相互排他ロックを取得するために pthread_atfork(3C) を呼び出すようにしてください。

スケジューリング

POSIX スレッド

Solaris は、次の 3 つの POSIX スケジューリングポリシーをサポートしています:

SCHED_OTHER
: 従来のタイムシェアリングスケジューリングポリシー。これは、タイムシェアリング (TS) スケジューリングクラスに基づいています。

SCHED_FIFO
: 先入れ先出しスケジューリングポリシー。このポリシーに従ってスケジュールされたスレッドは、より高い優先順位によって横取りされなければ、完了まで続行されます。このようなスレッドは、リアルタイム (RT) スケジューリングクラスに属します。呼び出し元プロセスの実効セットで {PRIV_PROC_PRIOCNTL} 特権が表明されている必要があります。

SCHED_RR　ラウンドロビンスケジューリングポリシー。このポリシーに従ってスケジュールされたスレッドは、より高い優先順位によって横取りされなければ、システムで決定された期間にわたって実行されます。このようなスレッドは、リアルタイム (RT) スケジューリングクラスに属し、呼び出し元プロセスの実効セットで {PRIV_PROC_PRIOCNTL} 特権が表明されている必要があります。

上の POSIX で指定されたスケジューリングポリシーに加えて、Solaris では次のスケジューリングポリシーもサポートされています:

SCHED_IA　スレッドは、priocntl(2) で説明されている対話型クラス (IA) ポリシーに従ってスケジュールされます。

SCHED_FSS　スレッドは、priocntl(2) で説明されている公平配分クラス (FSS) ポリシーに従ってスケジュールされます。

SCHED_FX　スレッドは、priocntl(2) で説明されている固定優先順位クラス (FX) ポリシーに従ってスケジュールされます。

Solaris スレッド　サポートされている唯一のスケジューリングポリシーが、TS スケジューリングクラスに基づいた SCHED_OTHER (タイムシェアリング) です。

エラー

マルチスレッドアプリケーションでは、別のスレッドが forkall(2) を呼び出したとき、ブロックしているシステムコールから EINTR が返されることがあります。

使用法

-mt コンパイラオプション　-mt コンパイラオプションは、マルチスレッド対応コードのためのコンパイルとリンクを行います。このオプションは –D_REENTRANT でソースファイルをコンパイルし、一連のサポートライブラリを正しく拡張します。

属性

属性についての詳細は、マニュアルページの attributes(5) を参照してください。

属性タイプ	属性値
MT レベル	MT-安全、Fork 1-安全

関連項目

crle(1)、fork(2)、priocntl(2)、libpthread(3LIB)、librt(3LIB)、libthread(3LIB)、pthread_atfor

『リンカーとライブラリ』

名前

trusted_extensions – Solaris Trusted Extensions

機能説明

Solaris Trusted Extensions ソフトウェアは、Solaris オペレーティングシステム (Solaris OS) 構成の 1 つです。Solaris Trusted Extensions (Trusted Extensions) には、デスクトップシステム、ウィンドウシステム、ゾーンシステム、ファイルシステム、およびネットワーク通信用に、ローカルオブジェクトとローカルプロセスのラベルが提供されています。これらのラベルは、ラベルの関係に基づいて情報のフローを制限するマルチレベルセキュリティー (MLS) ポリシーの実装に使用されます。所有権に基づく随意アクセス制御 (Discretionary Access Control、DAC) とは異なり、Trusted Extensions によって適用される MLS ポリシーは必須アクセス制御 (Mandatory Access Control、MAC) の一例です。

デフォルトでは、Trusted Extensions ソフトウェアは無効になっています。これは、次の FMRI で識別される `labeld(1M)` サービスによって有効化または無効化されます (ただし、構成はされません)。

```
svc:/system/labeld:default
```

Trusted Extensions ソフトウェアを使用する前に必要な構成については、下記の管理者ガイドを参照してください。Trusted Extensions ソフトウェアをアクティブ化または非アクティブ化するには、`labeld` を有効または無効にしたあとでシステムをリブートする必要があります。

関連項目

`labeld(1M)`, `label_encodings(4)`, `labels(5)`

『Trusted Extensions 構成と管理』

『Trusted Extensions ユーザーズガイド』

名前 vgrindefs – vgrind's language definition data base

形式 `/usr/lib/vgrindefs`

機能説明 `vgrindefs` contains all language definitions for `vgrind(1)`. Capabilities in `vgrindefs` are of two types: Boolean capabilities which indicate that the language has some particular feature and string capabilities which give a regular expression or keyword list. Entries may continue onto multiple lines by giving a \ as the last character of a line. Lines starting with # are comments.

Capabilities The following table names and describes each capability.

Name	Type	Description
ab	str	Regular expression for the start of an alternate form comment
ae	str	Regular expression for the end of an alternate form comment
bb	str	Regular expression for the start of a block
be	str	Regular expression for the end of a lexical block
cb	str	Regular expression for the start of a comment
ce	str	Regular expression for the end of a comment
id	str	String giving characters other than letters and digits that may legally occur in identifiers (default '_')
kw	str	A list of keywords separated by spaces
lb	str	Regular expression for the start of a character constant
le	str	Regular expression for the end of a character constant
oc	bool	Present means upper and lower case are equivalent
pb	str	Regular expression for start of a procedure
pl	bool	Procedure definitions are constrained to the lexical level matched by the 'px' capability
px	str	A match for this regular expression indicates that procedure definitions may occur at the next lexical level. Useful for lisp-like languages in which procedure definitions occur as subexpressions of defuns.
sb	str	Regular expression for the start of a string
se	str	Regular expression for the end of a string
tc	str	Use the named entry as a continuation of this one
tl	bool	Present means procedures are only defined at the top lexical level

Regular Expressions

`vgrindefs` uses regular expressions similar to those of `ex(1)` and `lex(1)`. The characters '^', '$', ':', and '\' are reserved characters and must be 'quoted' with a preceding \ if they are to be included as normal characters. The metasymbols and their meanings are:

`$`	The end of a line
`^`	The beginning of a line
`\d`	A delimiter (space, tab, newline, start of line)
`\a`	Matches any string of symbols (like '.*' in lex)
`\p`	Matches any identifier. In a procedure definition (the 'pb' capability) the string that matches this symbol is used as the procedure name.
`()`	Grouping
`|`	Alternation
`?`	Last item is optional
`\e`	Preceding any string means that the string will not match an input string if the input string is preceded by an escape character (\). This is typically used for languages (like C) that can include the string delimiter in a string by escaping it.

Unlike other regular expressions in the system, these match words and not characters. Hence something like '(tramp|steamer)flies?' would match 'tramp', 'steamer', 'trampflies', or 'steamerflies'. Contrary to some forms of regular expressions, `vgrindef` alternation binds very tightly. Grouping parentheses are likely to be necessary in expressions involving alternation.

Keyword List

The keyword list is just a list of keywords in the language separated by spaces. If the 'oc' boolean is specified, indicating that upper and lower case are equivalent, then all the keywords should be specified in lower case.

使用例

例 1 A sample program.

The following entry, which describes the C language, is typical of a language entry.

```
C|c|the C programming language:\
    :pb=^\d?*?\d?\p\d?(\a?\)(\d|{):bb={:be=}:cb=/*:ce=*/:sb=":se=\e":\
    :le=\e':tl:\
    :kw=asm auto break case char continue default do double else enum\
    extern float for fortran goto if int long register return short\
    sizeof static struct switch typedef union unsigned void while #define\
    #else #endif #if #ifdef #ifndef #include #undef # define endif\
    ifdef ifndef include undef defined:
```

Note that the first field is just the language name (and any variants of it). Thus the C language could be specified to `vgrind(1)` as 'c' or 'C'.

ファイル　　`/usr/lib/vgrindefs`　file containing vgrind descriptions

関連項目　　`ex(1)`, `lex(1)`, `troff(1)`, `vgrind(1)`

名前

wbem – Web-Based Enterprise Management

機能説明

Web-Based Enterprise Management (WBEM) is a set of management and Internet-related technologies intended to unify the management of enterprise computing environments. Developed by the Distributed Management Task Force (DMTF), WBEM enables organizations to deliver an integrated set of standards-based management tools that support and promote World Wide Web technology. The DMTF has developed a set of standards that make up WBEM. This set of standards includes:

Common Information Model (CIM)

CIM is an object-oriented data model that describes the overall management of information in an enterprise network environment. CIM consists of a CIM specification and a CIM schema:

CIM Specification — Consists of the language and methodology that describes management data.

CIM Schema — Provides actual model descriptions of systems, applications, large area networks, and devices. The CIM Schema enables applications from different developers on different platforms to describe management data in a standard format. As a result, a variety of management applications can share this information.

CIM Operations Over HyperText Transport Protocol (HTTP) 1.1 is a transport mechanism that maps CIM operations to HTTP to allow implementations of CIM to interoperate in an open, standardized manner.

CIM Operations Over HTTP 1.1 uses eXtensible Markup Language (XML), which is a markup language that represents management information in textual form.

In addition to the XML representation, CIM information is also represented textually by the managed object format (MOF). These MOF representations are typically stored as text files that developers compile into a CIM Object Manager.

WBEM Tools and Services

Tools and services that enable developers to create and Services management applications and instrumentation that manage heterogeneous computer environments include:

- Solaris WBEM Services
- Solaris WBEM Software Development Kit (SDK)

Solaris WBEM Services

These services consist of a set of value-added Services components. These services make it easier for developers to create management applications that run in the Solaris operating environment. They also make the Solaris operating environment easier to manage. Solaris WBEM Services consists of:

- CIM Object Manager, CIM Repository, and MOF Compiler

- CIM and Solaris Schema, which is an extension schema of CIM. CIM and Solaris Schema is a collection of CIM classes that describe managed elements in the Solaris operating environment. These classes are available from the CIM Object Manager at start up.
- Solaris Providers, which are programs that communicate information between the Solaris operating environment and the CIM Object Manager (providers get and set dynamic information about managed elements, acting as an intermediary between the CIM Object Manager and the managed elements).

  Solaris software providers have been developed for a variety of areas: users, roles, file systems, and network configuration, for example. A remote provider is also available to distribute agents away from the CIM Object Manager when required. Because of the incremental development capabilities of the WBEM instrumentation framework, developers can progressively and consistently add more providers for additional Solaris software services.
- SNMP Adapter for WBEM, which enables Simple Network Management Protocol (SNMP) management applications to access system management information that is provided by Solaris WBEM Services. Used with the Solstice Enterprise Agent (SEA) Master Agent `snmpdx(1M)`, the SNMP Adapter for WBEM maps SNMP requests into equivalent WBEM Common Information Model (CIM) properties or instances.

  The SNMP Adapter for WBEM also remaps the response from the CIM Object Manager into an SNMP response, which is returned to the management application.

  A mapping file contains the corresponding Object Identifier (OID), class name, property name, and Abstract Syntax Notation One (ASN.1) type for each object. Developers can create their own mapping files.
- SNMP Provider, which enables WBEM services to deliver SNMP information.

Solaris WBEM SDK

The Solaris WBEM SDK is a set of application programming interfaces (APIs) that contain the components necessary to write management applications. These applications communicate with WBEM-enabled management devices by using XML and HTTP communication standards.

Solaris WBEM applications request information or services from the Common Information Model (CIM) Object Manager through the WBEM APIs. These APIs represent CIM objects as Java classes. The APIs are used to describe managed objects and to retrieve information about managed objects in a system environment. The advantage of modeling managed resources by using CIM is that those objects can be shared across any system that is CIM-compliant.

For more information on the Solaris WBEM SDK, see the Solaris WBEM Developer's Guide. The Solaris WBEM API documentation is available in Javadoc format with the Solaris OS installation at `/usr/sadm/lib/wbem/doc/index.html`.

Compatibility of Solaris WBEM Services with Existing Protocols

Adapters and converters enable Solaris WBEM Services of Solaris to work compatibly with existing protocols by mapping WBEM information to these protocols. One such protocol is Simple Network Management Protocol (SNMP).

Legacy management applications can administer WBEM-enabled software in the Solaris operating environment. Developers can write agents or providers that convert information from these protocols to WBEM, and they can write adapters that convert WBEM information into these protocols.

属性

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SPARC and x86
Architecture	SUNWwbapi, SUNWwbco, SUNWwbco, SUNWwbdev, SUNWwbdoc, SUNWwbpro
CSI	Enabled

関連項目

appletviewer(1), cimworkshop(1M), init.wbem(1M), mofcomp(1M), mofreg(1M), snmpdx(1M), wbemadmin(1M), wbemconfig(1M), wbemlogviewer(1M), attributes(5)

名前

zones – Solaris アプリケーションコンテナ

機能説明

Solaris のゾーン機能を使用して、アプリケーションを実行するための独立した環境を作成することができます。ゾーンで実行するプロセスは、システム上のほかのアクティビティーから監視されたり干渉されたりすることはありません。ほかのプロセス、ネットワークインタフェース、ファイルシステム、デバイス、およびプロセス間通信機能へのアクセスは、別のゾーンで実行されているプロセスと対話することがないように制限されます。

ゾーンで使用できる特権は、操作がシステム全体に影響することがないように制限されます。privileges(5) を参照してください。

ゾーンの構成と管理は、zoneadm(1M) および zonecfg(1M) ユーティリティーを使って行います。ゾーンごとに詳細な構成を指定し、ファイルシステムの内容 (ソフトウェアパッケージなど) をゾーンにインストールし、ゾーンの実行時の状態を管理することができます。zlogin(1) を使用すれば、アクティブなゾーンでコマンドを実行することができます。この操作を行うときに、in.rlogind(1M) や sshd(1M) などのネットワークベースのログインサーバーからログインする必要はありません。

ゾーンを自動的にブートするかどうかは、次のように FMRI で識別されるゾーンサービスを使って切り替えます。

svc:/system/zones:default

zoneadm(1M) を参照してください。ゾーンの autoboot プロパティーに true を設定して、常に自動的にブートする方法もあります。ただし、ゾーンサービスが無効になっている場合は、ゾーンの autoboot プロパティーの設定に関係なく、ゾーンは自動的にブートしません。zonecfg(1M) を参照してください。

アクティブなゾーンは、英数字の名前と数値の ID によって識別されます。英数字の名前は、zonecfg(1M) ユーティリティーを使用して構成します。数値の ID は、ゾーンがブートするときに自動的に割り当てられます。zonename(1) ユーティリティーを実行すると、現在のゾーン名が報告されます。zoneadm(1M) ユーティリティーを実行すると、構成済みゾーンの名前と ID が報告されます。

ゾーンの状態には、次のいくつかの種類があります。

構成済み
: ゾーンの構成が完全に指定および確定されて、安定した記憶領域に保存されていることを示します。

不完全
: ゾーンのインストールまたはアンインストールが完了していないか、またはそのような状態への移行中に操作が中断されたことを示します。

インストール済み
: ゾーンの構成がシステムにインストールされたことを示します。 パッケージはゾーンのルートパスにインストールされています。

準備完了
ゾーンの仮想プラットフォームが確立されていることを示します。たとえば、ファイルシステムのマウントおよびデバイスの構成は完了していますが、ゾーンに関連付けられているプロセスはまだ開始していません。

稼働
ゾーンのアプリケーション環境に関連付けられたユーザープロセスが稼働状態であることを示します。

停止処理中
停止
ゾーンが停止していることを示します。アプリケーション環境の状態 (ファイルシステムをマウントした状態など) を切り離せない場合、または仮想プラットフォームの一部を破棄できない場合に、ゾーンがそれらの状態に固定されることがあります。そのような場合には、オペレータの介入が必要です。

UNAVAILABLE
ゾーンがインストールされたが、ブートできないことを示します。ゾーンが利用不可能な状態になるのは、`svc:/system/zones:default` がオンラインになろうとしているときや、ゾーンがブートしようとしているときにそのゾーンのストレージが使用できないとき、アーカイブの抽出が成功したあとにアーカイブベースのインストールが失敗したとき、および (正しくない強制された接続のあとなど) そのゾーンのソフトウェアに大域ゾーンのソフトウェアとの互換性がないときです。

プロセスのアクセス制限

あるゾーン (大域ゾーンを除く) で実行中のプロセスからほかのプロセスへのアクセスは制限されています。`/proc` (`proc(4)` を参照) またはプロセス ID を引数に持つシステムコールインタフェース (`kill(2)`、`priocntl(2)` など) を使用してアクセスできるプロセスは、同じゾーンのプロセスだけです。ほかのゾーン (大域ゾーンを含む) にあるプロセスにアクセスしようとすると、指定したプロセスがなかった場合に返されるエラーコードと同じエラーコードで失敗します。

特権の制限

非大域ゾーンで実行されるプロセスでは、あるゾーンで実行する操作がほかのゾーンの操作に影響することがないように、一部の特権が制限されています。特権が制限されることによって、そのゾーンで特権ユーザー (スーパーユーザー、ルートユーザーなど) が実行できる操作が制限されます。ゾーンで許可される特権のリストは、`ppriv(1)` ユーティリティーを使って表示できます。特権に関する詳細については、`privileges(5)` を参照してください。

デバイスの制限

あるゾーンのプロセスが別のゾーンのプロセスに干渉することがないよう、ゾーン内部で利用可能なデバイスセットには制限があります。たとえば、ゾーンで実行されるプロセスから、`/dev/kmem` を使用してカーネルメモリーを変更したり、ルートディスクの内容を変更したりすることはできません。このため、デフォルトでは、ゾーン内で安全に利用可能であると見なされるいくつかの仮想デバイスだけを使用できます。特定のゾーンで使用できるデバイスを追加するときは、`zonecfg(1M)` ユーティリティーを使用します。

デバイスおよび特権が制限されることで、非大域ゾーンで実行できるユーティリティーにもいくつかの影響があります。たとえば、eeprom(1M)、prtdiag(1M)、および prtconf(1M) ユーティリティーは、通常は利用できないデバイスを操作するため、ゾーンでは使用できません。

ブランド

ゾーンを最初に作成するとき、ゾーンにブランドを割り当てることができます。ブランドゾーン内のソフトウェアは、大域ゾーンにある同じソフトウェアとは異なります。このようなソフトウェアには、構成や配置の異なる Solaris ソフトウェア、Solaris 以外のソフトウェアなどがあります。このような特定のソフトウェアの集まりを「ブランド」と呼びます (brands(5) を参照)。いったんインストールされると、ゾーンをアンインストールしない限り、ゾーンのブランドは変更できません。

ファイルシステム

各ゾーンは、ゾーンの root と呼ばれるディレクトリをルートとする、ファイルシステム階層の独自領域を保持します。ゾーンで実行されるプロセスがアクセスできるファイルは、階層のその領域にあるファイル (ゾーンのルートにあるファイル) だけです。この制限により、あるゾーンのプロセスから、別のゾーンに関連付けられているファイルシステムのデータを破壊または検査できないようになっています。chroot(1M) ユーティリティーをゾーンで使用することはできますが、その処理はゾーンでアクセスできるルートパスだけに制限されます。

ファイルシステムの領域を保護するために、ファイルシステムの領域を 1 つ以上のゾーンにマウントするときには、lofs(7FS) ファイルシステムの読み取り専用オプションを使用して行います。このように操作することで、同じファイルシステムのデータを複数のゾーンで共有するときに、ゾーンのセキュリティーを確保することができます。

あるゾーンで確立された NFS および autofs マウントへのアクセスは、そのゾーンだけに制限され、ほかのゾーン (大域ゾーンを含む) からアクセスすることはできません。ゾーンが停止またはリブートすると、マウントは削除されます。

ゾーンに委任された ZFS データセットは、そのゾーン内で管理できます。委任されたデータセット内に子データセットを作成できます。委任されたデータセット内にデータセットを作成すると、それらも委任されます。委任されたデータセットは、最上位のもの以外は破棄できます。委任されたデータセットには、ほとんどのプロパティーを設定できます。詳細については、zfs(1M) を参照してください。

各ゾーンには委任された最上位のデータセットが 1 つあり、そのデータセットには ROOT データセットがあるほか、 .../export や .../export/home などのデータセットが含まれる場合もあります。ROOT データセットの下にあるデータセットは、そのゾーンのブート環境を構成します。ブート環境のデータセットを作成または破棄する場合は、必ず zoneadm(1M) コマンドまたは beadm(1M) コマンドを使用するようにしてください。

ネットワーク処理

ゾーンでは、TCP、UDP、および SCTP アプリケーションに独自のポート番号空間を使用し、通常は 1 つ以上の個別の IP アドレスを使用します (ただし、Trusted Extensions の特定の構成ではゾーン間で IP アドレスが共有される)。

ゾーンでは、IP レイヤー (IP ルーティング、ARP、IPsec、IP フィルタなど) の構成および状態を大域ゾーンと共有するか (共有 IP ゾーン)、異なる IP レイヤーの構成および状態を使用する (排他的 IP ゾーン) ことができます。

ゾーンが大域ゾーンと同じデータリンクに接続される、つまり、同じ IP サブネット上に配置される場合は、ゾーンに共有 IP インスタンスを使用することをお勧めします。

ゾーンが大域ゾーンやほかの非大域ゾーンとは異なる VLAN や LAN に接続される場合など、ネットワークの IP レイヤーで隔離されている必要がある場合は、隔離の理由から、ゾーンに排他的 IP を使用するようにしてください。

共有 IP ゾーンでは、ネットワークに対する特定の操作 (IP アドレスの変更、IP または Ethernet の偽装パケットの送信など) が禁止されます。これに対し、排他的 IP ゾーンでは、同じネットワークインタフェースに接続されている独立したホストと同等のネットワーク機能を使用できます。特に、そのようなゾーンのスーパーユーザーは、IP アドレスの変更および ARP パケットの偽装を行うことができます。

共有 IP ゾーンには、1 つ以上のネットワークインタフェース名と IP アドレスが `zonecfg(1M)` で割り当てられます。ネットワークインタフェース名は大域ゾーンでも構成する必要があります。

排他的 IP ゾーンには、1 つ以上のネットワークインタフェース名が `zonecfg(1M)` で割り当てられます。ネットワークインタフェース名はそのゾーンに排他的に割り当てる必要があります。つまり、同じネットワークインタフェース名をほかの稼働中のゾーンに割り当てたり、大域ゾーンで使用したりすることはできません。

DHCP クライアント、IPsec および IP フィルタという形での IP レベルの完全な機能は、排他的 IP ゾーンで使用できますが、共有 IP ゾーンでは使用できません。

ホスト識別子

ゾーンは 32 ビットのホスト識別子をエミュレートできます。ホスト識別子は、システム統合を目的として、`zonecfg(1M)` 経由で構成できます。ゾーンがホスト識別子をエミュレートすると、`hostid(1)` や `sysdef(1M)` などのコマンドだけでなく、ゾーンのコンテキスト内で実行される `sysinfo(2)` や `gethostid(3C)` などの C インタフェースも、ホストマシンの識別子ではなく、エミュレートされたゾーンのホスト識別子を表示するか、返します。

属性

属性についての詳細は、マニュアルページの `attributes(5)` を参照してください。

属性タイプ	属性値
使用条件	system/core-os

関連項目

hostid(1), zlogin(1), zonename(1), beadm(1M), in.rlogind(1M), sshd(1M), sysdef(1M), zfs(1M), zoneadm(1M), zonecfg(1M), kill(2), priocntl(2), sysinfo(2), gethostid(3C), getzoneid(3C), ucred_get(3C), proc(4), attributes(5), brands(5), privileges(5), crgetzoneid(9F)